CITY UNIVERSITY OF HONG KONG

DEPARTMENT OF PHYSICS

BACHELOR OF SCIENCE IN APPLIED PHYSICS
2024-2025 PROJECT

Bragg and Anti-Bragg Band Gap with Superconducting Qubit

by

RANGDHOL Khushi Vishwanath (57194010)

November 2024

Submitted in partial fulfilment of the

requirements for the degree of

BACHELOR OF SCIENCE

IN

APPLIED PHYSICS

from

City University of Hong Kong

November 2024

Project Supervisor	Io Chun Hoi

Abstract

The following paper reports on the formation of the photonic band gaps in superconducting qubit arrays arranged in the Bragg and anti-Bragg configurations within the waveguide quantum electrodynamics framework. Particular superconducting qubits are used, such as Yttrium Iron Garnet resonators, due to their strong coherent electromagnetic coupling that makes this system one of the most promising candidates for control over quantum light-matter interactions. Our study combines theoretical modeling with computational simulations to investigate how qubit spacing and configuration can affect transmission and reflection properties. We use transfer matrix methods to simulate the transmission coefficient across a range of configurations and probe frequencies, paying attention to the interference patterns created by constructive and destructive interactions in Bragg and anti-Bragg structures. Results We find Bragg configurations provide large band gaps due to constructive interference which serves to block certain frequencies whereas anti-Bragg configurations provide transparency windows due to destructive interference. Contour plots of transmission coefficients exhibit clear band gap behaviour that underlines the flexibility of such structures in quantum photonic applications. This work contributes to the understanding of the engineering of photonic band gaps in quantum systems and has potential applications in photon routing, quantum memory, and the design of photonic crystals. Future efforts may continue to explore non-linear effects and multi-mode coupling in order to further extend this work into advanced quantum communication networks.

Acknowledgement

I am deeply grateful to my professor, Dr. Io Chun Hoi, for his invaluable guidance and unwavering support throughout the duration of this research project. His mastery and depth of understanding in relation to the subtleties of superconducting qubits, with particular emphasis on Bragg and anti-Bragg gaps, proved to be an indispensable tool in my development of full understanding of this highly intricate subject.

I would like to thank for the deeper understanding that this project has provided into the underlying principles that control the behaviour of superconducting qubits and the importance of photonic band gap structures. I have studied how Bragg and anti-Bragg gaps can serve to increase coherence times and control qubit interactions crucial to the advance of quantum computing technologies. This experience has expanded not only my knowledge but also a passion for the further exploration of quantum mechanics and its applications. Once again, I wish to express my heartfelt thanks to Dr. Io Chun Hoi, for his supervision and encouragement, which have turned this research work enlightening as well as rewarding.

Contents

1 Introduction

1.1 Background

Superconducting qubits are becoming essential components of quantum technologies, particularly in fields like quantum computing and quantum communication. These qubits have strong interactions with electromagnetic fields and can maintain their quantum state for extended periods, known as having long coherence times, which makes them ideal for manipulating light and matter at the quantum level [1]. Superconducting qubits leverage Josephson junctions, which enable them to operate as artificial atoms with discrete energy levels, offering a high degree of control and tunability [2].

One effective way to study the interactions between superconducting qubits and light is through waveguide quantum electrodynamics (wQED), where qubits are coupled to a continuous spectrum of electromagnetic modes in a waveguide. In this setup, qubits can strongly interact with confined light, leading to unique quantum effects such as photon-qubit interactions, resonance fluorescence, and the formation of band gaps, which prevent certain wavelengths of light from propagating through the system [3].

In this context, Bragg and anti-Bragg structures are of particular interest. Bragg structures are formed by arranging qubits at intervals that lead to constructive interference. This arrangement enhances reflection and forms band gaps by aligning the light waves from each qubit in such a way that they reinforce each other, blocking certain frequencies from passing through. On the other hand, anti-Bragg structures rely on destructive interference, where the waves from the qubits cancel each other out, allowing light to transmit through the waveguide more freely [4]. These structures provide powerful tools for controlling light propagation within qubit arrays, enabling applications like photon routers and quantum memories by selectively permitting or blocking light.

1.2 Objectives

The primary objective of this study is to investigate how Bragg and anti-Bragg structures can be used to generate and manipulate band gaps in superconducting qubit arrays. Specifically, we aim to:

- To investigate the Bragg and anti-Bragg configurations that would yield a photonic band gap generation in waveguide systems
- Explain how factors in qubit spacing, phase alignment, and coherence influence the transmission and reflection properties of qubit arrays
- Explore potential applications of such structures including selectivity for transmitting and reflecting light that can be used in service for advanced quantum devices, such as photon routers and quantum memories.

2 Literature review

2.1 Superconducting Qubits and Waveguide Quantum Electrodynamics (wQED)

Superconducting qubits represent one of the most intensively investigated systems owing to the strong coupling between them and the electromagnetic fields, as well as long coherence times suitable for quantum technologies like computing and communication. In waveguide quantum electrodynamics (wQED), qubits are coupled to a waveguide that supports continuous electromagnetic modes. This stands in contrast to cavity QED, where atoms interact with only a single mode in a confined cavity. In wQED, qubits can interact with a wider range of photon modes, enhancing the overall interaction strength [2].

The Hamiltonian for a single qubit interacting with the waveguide is:

$$H = \frac{\omega_q}{2}\sigma_z + \int_{-\infty}^{\infty} \omega_k a_k^\dagger a_k \, dk + \int_{-\infty}^{\infty} g_k \left(\sigma_+ a_k + \sigma_- a_k^\dagger\right) dk$$

where ω_g is the qubit's frequency, σ_z is the Pauli operator, and g_k is the coupling strength between the qubit and the waveguide mode k, and $a_k^\dagger$, a_k are the photon creation and annihilation operators [5]. This coupling gives rise to phenomena such as resonance fluorescence, where the qubit absorbs and re-emits photons, and photon-qubit coupling, which leads to interference effects in the waveguide.

Compared to cavity quantum electrodynamics, waveguide QED offers spacious ground for studies on quantum light-matter interactions, as in general, interactions with continuous modes may take part over a much larger frequency range. These interactions are the basis for Bragg and anti-Bragg structures that occur in qubit arrays and correspondingly have critical interference effects in the control of photon propagations [3].

2.2 YIG Spheres and Magnonic Modes as Alternatives to Qubits in wQED

Yttrium Iron Garnet (YIG) spheres have recently emerged as a powerful alternative to traditional qubits for wQED experiments. Unlike qubits, which typically operate as two-level systems, YIG spheres exhibit multi-level energy states because of their unique magnetic properties. In YIG spheres, magnons, which are quantized spin-wave excitations or collective oscillations of electron spins can interact with photons. This photon-magnon coupling happens in the microwave frequency range, making YIG ideal for wQED experiments that need stable quantum states and efficient coupling to light [11].

Key properties of YIG spheres that make them ideal for wQED applications include:

– **High Coherence Time**: Like superconducting qubits, YIG spheres can maintain quantum states for long periods. This stability makes them well-suited for experiments where precise control over quantum states is crucial.

– **Strong Photon-Magnon Coupling**: The strong interaction between photons and magnons allows energy to move efficiently between them, enabling the study of light-matter interactions. This interaction can happen in both single-photon and multi-photon (low-power) regimes, which makes YIG suitable for a wide range of wQED experiments [12].

– **Magnetic Tunability**: The resonance frequency of magnons in YIG spheres can be adjusted by applying an external magnetic field. This allows experimenters to control the coupling frequency precisely, making it easy to align the magnon's frequency with that of the waveguide photons [11].

– **Low Damping Rates**: YIG spheres have very low magnetic damping, meaning their magnons can oscillate for longer times without losing much energy. This is important for maintaining coherent photon-magnon interactions over extended durations, contributing to the stability of the experimental system.

These properties make YIG spheres an excellent choice for studying wQED in both low-power and multi-photon regimes. By utilizing YIG's multi-level structure and stable magnon modes, researchers can explore photon-magnon interactions in greater depth than with traditional two-level qubits.

2.3 Bragg and Anti-Bragg Structures in Qubit Analysis

In qubit arrays, Bragg structures arise when qubits are arranged periodically, causing constructive interference between the light scattered by the qubits. This leads to the formation of band gaps, regions where certain wavelengths of light are blocked from passing through. The Bragg condition for maximum reflection is given by:

$$d = \frac{n\lambda}{2}$$

where d is the distance between qubits, λ is the wavelength of the light, and n is an integer [6]. This periodic spacing creates a band gap by reflecting photons of specific wavelengths, similar to how atomic mirrors reflect light.

In contrast, anti-Bragg structures rely on destructive interference. Here, the qubits are spaced such that the light scattered by each qubit cancels out, allowing more light to pass through the waveguide. Anti-Bragg structures create transparency windows within the band gap, enhancing transmission rather than reflection [4]. The ability to control whether a qubit array forms a Bragg or anti-Bragg structure depends on the spacing phase alignment, and coherence of the qubits.

2.4 Applications of Bragg and Anti-Bragg Gaps

Bragg and anti-Bragg gaps are very important in electromagnetic wave propagation manipulation and thus are valued in a variety of applications in both photonics and quantum technologies. Bragg gaps are used to create photonic band gaps, where certain frequencies of light are blocked, enabling devices like photonic crystals, optical mirrors, and photon routers. Spacing the qubits or resonators at Bragg intervals produces constructive interference that forbids the transmission of certain frequencies, and hence gives highly reflective structures for isolating particular light modes or for the protection of quantum channels against signal loss [8].

While Bragg gaps act by prohibiting transmission through the action of destructive interference, anti-Bragg gaps, on the other hand, allow selective transmission. This property of anti-Bragg gaps, allowing controlled transparency windows, is very important in the design of optical filters and quantum memories. Quantum photonic circuits can implement anti-Bragg structures enabling frequency selection of qubits, effectively allowing regions where the transmission is optimized for specific wavelengths. This type of filtering is essential in quantum networks for distinguishing different quantum states or information carriers for independent processing [9].

Bragg and anti-Bragg configurations therefore form a versatile tool in photonic quantum network engineering, ranging from high reflectivity mirrors and optical isolators, over selective transmission filters to quantum routers. The engineering of band gaps and transparency windows forms plenty of options in the development of quantum systems operating sensitive control of light propagation-from quantum information processing to secure communications and photonic processing applications. [10].

3 Methodology

To investigate Bragg and Anti-Bragg band gap formation in superconducting qubit arrays, we analyzed transmission properties in the low-power regime, combining waveguide quantum electrodynamics (wQED) techniques with transfer matrix methods. This methodology details the steps taken to model and examine how various qubit configurations influence photonic transmission and band gap characteristics. Employing computational modelling and numerical simulations, we focused on capturing the effects of resonator arrangements on transmission behaviour. The methodology for this study is outlined as follows:

3.1 Experimental Setup

- **Waveguide and Qubit Arrangement**: The experiment uses a waveguide structure with YIG resonators positioned at specified intervals along its length. Each YIG resonator acts as a qubit, and the spacing between these resonators is varied to create either Bragg or anti-Bragg configurations. In a Bragg configuration, YIG resonators are spaced at intervals of approximately $\frac{d}{\lambda} = 0.5$, promoting constructive interference and reflection. For anti-Bragg configurations, spacings around $\frac{d}{\lambda} = 0.25$ and $\frac{d}{\lambda} = 0.75$ are used to encourage destructive interference, allowing selective transmission of certain frequencies through the waveguide.

- **Frequency Sweep and Transmission Measurements**: The probe frequency ω_p is swept across a range from 2.92 GHz to 3 GHz to observe how the YIG array's configuration affects transmission properties. Using low-power microwave signals, each frequency within the sweep is sent through the waveguide to interact with the YIG resonators. By measuring the transmitted and reflected signal amplitudes at each frequency, the setup enables analysis of the transmission coefficient $|S_{21}|$, allowing identification of band gaps or transparency windows based on qubit spacing.

- **Magnetic Conditions**: A static magnetic field is applied to each YIG qubit to tune its resonance frequency, aligning it with the probe microwave frequencies. This magnetic tunability is crucial for achieving controlled coupling between qubits and photons within the wQED system, and it allows for precise control over the qubit resonator states.

- **Data Collection and Contour Plot Analysis**: Transmission data is collected and processed for each YIG configuration (e.g., 2, 3, 4, 5, 6, and 10 YIGs) at different normalized distances. For analysis, transmission coefficients are mapped onto contour plots as a function of both probe frequency ω_p and normalized distance $\frac{d}{\lambda}$. These contour plots reveal the formation of photonic band gaps in Bragg structures and transparency windows in anti-Bragg structures, illustrating how the qubit spacing affects light transmission and band gap behavior in the wQED system.

We now explore the methods used in detail.

3.2 Transmission and Reflection Coefficients

To mathematically capture the transmission characteristics, we define the transmission coefficient $|S_{21}|$ and reflection coefficient $|S_{11}|$. In this context, the transmission coefficient describes how much of the input signal passes through the array, while the reflection coefficient measures how much is reflected back.

- **Transmission Coefficient** $|S_{21}|$

  The modified function for the transmission coefficient is defined as follows:

  $$S_{21} = 1 - \frac{\gamma_{\text{res}}}{2(\gamma_{\text{int}} + i(\omega - \omega_{\text{resonator}}))}$$

  where:

  - ω is the probe frequency.

- γ_{res} represents the decay rate due to coupling between the resonator and the waveguide.
- γ_{int} denotes the internal decay rate of the resonator.
- $\omega_{\text{resonator}}$ is the resonator's intrinsic frequency.

– **Reflection Coefficient** $|S_{11}|$

The reflection coefficient $|S_{11}|$ is derived from the transmission coefficient:

$$S_{11} = 1 - S_{21}$$

The reflection coefficient is particularly useful for examining resonance shifts and damping behavior in response to probe frequencies and resonator arrangements.

3.3 Transfer Matrix Calculation for Each Resonator

The transmission characteristics of each qubit resonator in the array were modelled using a transfer matrix T which describes the reflection and transmission properties of each qubit. The transfer matrix for each qubit is defined as:

$$T = \begin{pmatrix} \frac{1+2r}{1+r} & \frac{r}{1+r} \\ \frac{-r}{1+r} & \frac{1}{1+r} \end{pmatrix}$$

with components:

$$T_{11} = \frac{1+2r}{1+r}, \quad T_{12} = \frac{r}{1+r}, \quad T_{21} = \frac{-r}{1+r}, \quad T_{22} = \frac{1}{1+r}$$

where r is the reflection coefficient, influencing the transmission properties of the qubit. This approach was applied to each qubit in the array.

3.4 Propagation Phase Calculation

To account for the phase shift between adjacent qubits, a propagation phase matrix P was calculated for each distance and probe frequency. This phase matrix captures the distance-dependent phase delay essential for Bragg and Anti-Bragg interference.

The propagation phase matrix is represented as:

$$P = \begin{pmatrix} e^{i\phi} & 0 \\ 0 & e^{-i\phi} \end{pmatrix}$$

where the phase shift ϕ is defined as:

$$\phi = \frac{2\pi\omega_p \times d}{0.7\,c}$$

where ω_p is the probe frequency, d is the distance between qubits, and c is the speed of light. The factor 0.7 scales the phase shift based on specific experimental conditions.

3.5 Total Transfer Matrix for Multi-Resonator Configurations

For a system with multiple resonators, the total transfer matrix T_{total} is formed by sequentially multiplying each qubit's transfer matrix and the propagation phase matrix. For example, in a configuration with 5 qubits, the total transfer matrix is represented as:

$$T_{\text{total}} = T_1 \cdot P_1 \cdot T_2 \cdot P_2 \cdot T_3 \cdot P_3 \cdot T_4 \cdot P_4 \cdot T_5$$

This sequence can be extended to any number of resonators by alternating transfer matrices T_i with propagation phase matrices P_i.

3.6 Transmission Coefficient Calculation

The transmission coefficient $|S_{21}|^2$ for the entire resonator array is derived from the T_{22} element of the final transfer matrix T_{total}:

$$|S_{21}|^2 = \left|\frac{1}{T_{\text{total},22}}\right|^2$$

This squared magnitude of S_{21} enhances visualization of resonance effects and band gaps within the multi-resonator system.

3.7 Normalized Distance Calculation

To interpret the spacing between qubits in terms of wavelength, we define a normalized distance d_{norm} as:

$$d_{\text{norm}} = \frac{d \times 2.9623 \times 10^9}{0.7\,c}$$

where d represents the physical distance between qubits. This normalized distance allows us to express spacing in terms of wavelength, making it easier to analyze and compare transmission properties across configurations.

4 Results and Discussion

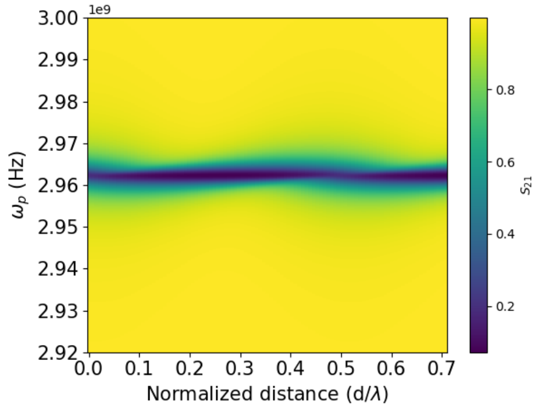
In this study, we have investigated the transmission properties for both Bragg and anti-Bragg structures in arrays of superconducting qubits based on the wQED framework. Using the transfer matrix method, we have studied how the arrangement of qubits affects the band gap formation in the low-power regime. In the following results section, we identify the output of our analysis, which is the behaviour of the transmission coefficients for a wide range of probe frequencies and different qubit configurations.

4.1 Transmission Characteristics of Different YIG Configurations

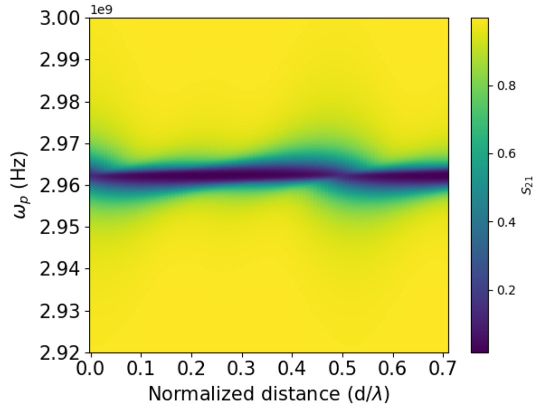
The transmission coefficient, given by $|S_{21}|$, was computed for different YIG resonator configurations (2 to 10 YIGs) using the transfer matrix approach [1]. As shown in (Figure 1), each additional YIG resonator contributed to a deeper transmission dip around the central frequency $\omega_p = 2.96GHz$. This effect corresponds to increased destructive interference among the resonators, characteristic of Bragg structures, which suppresses transmission near resonance.

4.2 Contour Analysis of $|S_{21}|$ with Normalized Distance

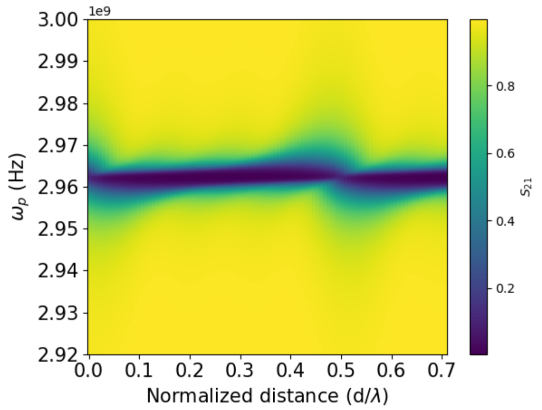
To visualize the transmission as a function of both probe frequency and normalized distance, contour plots were generated for configurations of 2, 3, 4, 5, 6, and 10 YIGs (Figure 1). These plots reveal the formation of band gaps and highlight how different resonator numbers affect the spectral position and



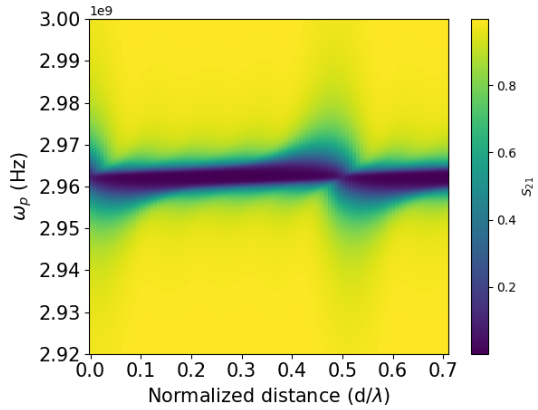
(a) Contour plot for transmission coefficient $|S_{21}|$ as a function of normalized distance $\frac{d}{\lambda}$ and probe frequency for 2 YIG resonators



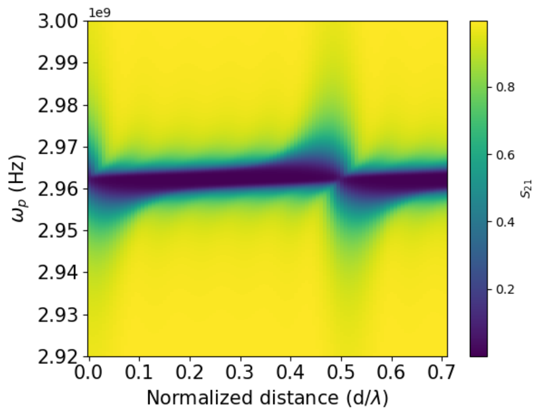
(b) Contour plot for transmission coefficient $|S_{21}|$ as a function of normalized distance $\frac{d}{\lambda}$ and probe frequency for 3 YIG resonators



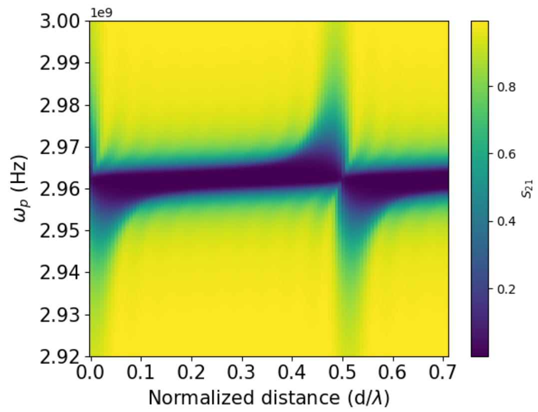
(c) Contour plot for transmission coefficient $|S_{21}|$ as a function of normalized distance $\frac{d}{\lambda}$ and probe frequency for 4 YIG resonators



(d) Contour plot for transmission coefficient $|S_{21}|$ as a function of normalized distance $\frac{d}{\lambda}$ and probe frequency for 5 YIG resonators



(e) Contour plot for transmission coefficient $|S_{21}|$ as a function of normalized distance $\frac{d}{\lambda}$ and probe frequency for 6 YIG resonators



(f) Contour plot for transmission coefficient $|S_{21}|$ as a function of normalized distance $\frac{d}{\lambda}$ and probe frequency for 10 YIG resonators

Figure 1: Contour plots of the transmission coefficient $|S_{21}|$ of YIG resonators over a range of distances and frequencies. Contains the transmission characteristics for each distance frequency pair

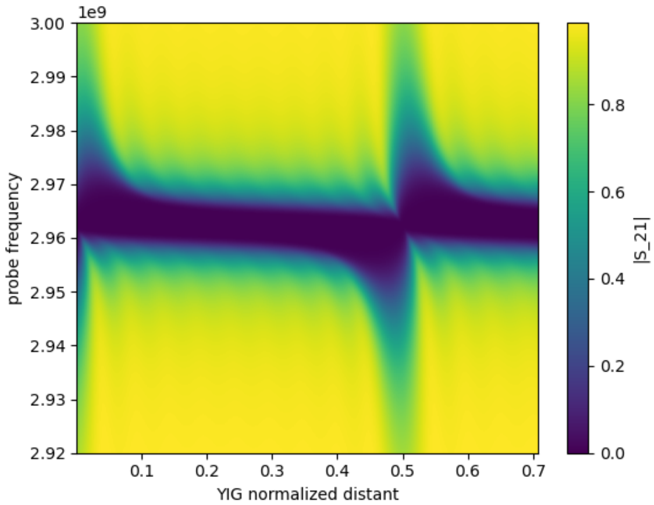


Figure 2: Contour plot of the transmission coefficient $|S_{21}|$as a function of normalized distance $\frac{d}{\lambda}$ and probe frequency for a 10 YIG resonator configuration, using fine steps in normalized distance. This plot highlights the detailed transmission behavior and band gap formation across a broad range of distances and frequencies.

width of the band gaps. The contour plots show that as the number of YIG resonators increases, the transmission suppression region (band gap) becomes more prominent, consistent with predictions of Bragg and anti-Bragg band gap formations [2].

4.3 Effect of Normalized Distance on Bragg and Anti-Bragg Structures

The normalized distance $\frac{d}{\lambda}$ between qubits influenced the width and position of the band gaps. For distances around $\frac{d}{\lambda} \approx 0.5$, typical of Bragg-like conditions, transmission was strongly suppressed, forming a wide band gap around the central frequency. This behavior aligns with theoretical predictions that Bragg configurations should exhibit maximum reflection and minimal transmission near resonance due to constructive interference [3].

In contrast, anti-Bragg conditions, such as those at normalized distances of $\frac{d}{\lambda} = 0.5$ and $\frac{d}{\lambda} = 0.75$, allow for destructive interference in the reflected waves. This interference cancellation enables transmission peaks in regions where Bragg structures suppress transmission, resulting in narrow transparency windows. These findings indicate that anti-Bragg configurations can allow selective transmission in specific frequency bands, creating transparency windows within band gap regions. Contour plots at these intervals show distinct transmission peaks, illustrating how anti-Bragg spacing reduces the reflected wave buildup, effectively "opening" the band gaps.

5 Conclusion

In this work, we theoretically and computationally explored the Bragg and anti-Bragg band gaps of superconducting qubit arrays, based on waveguide quantum electrodynamics. We show that light transmission properties can be engineered through constructive and destructive interference due to the precise spacing of YIG resonators within a waveguide. Specifically, Bragg structures formed through qubit spacing at approximately $d = \frac{\lambda}{2}$ created strong reflection regions, or band gaps, that blocked certain frequencies, while anti-Bragg structures at $d = \frac{\lambda}{4}$ and $d = \frac{3\lambda}{4}$ promoted transparency windows by allowing selective frequencies to pass through.

Using simulations, we showed that qubit spacing, phase alignment, and their coupling properties are very important for the modification of transmission and reflection characteristics in these systems. Such clear differences from the contour plots and transmission graphs between the Bragg and anti-Bragg configuration again proved the versatility of qubit spacing as a tool for shaping photonic band structures. These findings can therefore provide the basis for the design of quantum photonic devices with highly tunable transmission properties.

These results can be further extended in the future by considering more complex configurations such as multi-mode coupling effects and nonlinear interactions in order to extend functionalities of the wQED systems even further. Experimental verification of simulated band structures can also increase the applicability of Bragg and anti-Bragg structures in realistic quantum networks and photonic circuitry.

6 References

1. Gu, X., Kockum, A. F., Miranowicz, A., Liu, Y. X., Nori, F. (2017). "Microwave photonics with superconducting quantum circuits." Physics Reports, 718, 1-102.

2. Lalumière, K., Sanders, B. C., van Loo, A. F., Fedorov, A., Wallraff, A., Blais, A. (2013). "Input-output theory for waveguide QED with an ensemble of inhomogeneous atoms." Physical Review A, 88(4), 043806.

3. Shahmoon, E., Wild, D. S., Lukin, M. D., Yelin, S. F. (2017). "Cooperative resonances in light scattering from two-dimensional atomic arrays." Physical Review Letters, 118(11), 113601.

4. Hoi, I. C., Kockum, A. F., Palomaki, T., Stace, T. M., Fan, B., Tornberg, L., Wilson, C. M. (2013). "Giant cross-Kerr effect for propagating microwaves induced by an artificial atom." Physical Review Letters, 111(5), 053601.

5. Astafiev, O., Zagoskin, A. M., Abdumalikov Jr., A. A., Pashkin, Y. A., Yamamoto, T., Inomata, K., Nakamura, Y. (2010). "Resonance fluorescence of a single artificial atom." Science, 327(5967), 840-843.

6. Fan, S., Kocabas, S. E., Shen, J. T. (2010). "Input-output formalism for few-photon transport in one-dimensional nanophotonic waveguides coupled to a qubit." Physical Review A, 82(6), 063821.

7. Clarke, J., Wilhelm, F. K. (2008). "Superconducting quantum bits." Nature, 453(7198), 1031–1042.

8. Sakoda, K. (2005). "Optical Properties of Photonic Crystals." Springer Series in Optical Sciences.

9. Yablonovitch, E. (1987). "Inhibited Spontaneous Emission in Solid-State Physics and Electronics." Physical Review Letters, 58(20), 2059-2062.

10. Joannopoulos, J. D., Johnson, S. G., Winn, J. N., Meade, R. D. (2008). "Photonic Crystals: Molding the Flow of Light." Princeton University Press.

11. Rao, J. (2021). Design of cavity magnonic devices based on the dynamics of cavity magnon polaritons.

12. M Harder, Y Yang, BM Yao, CH Yu, JW Rao, YS Gui, RL Stamps, CM Hu Physical review letters 121 (13), 137203

Appendix

Transmission and Reflection Coefficients at Different Probe Frequencies

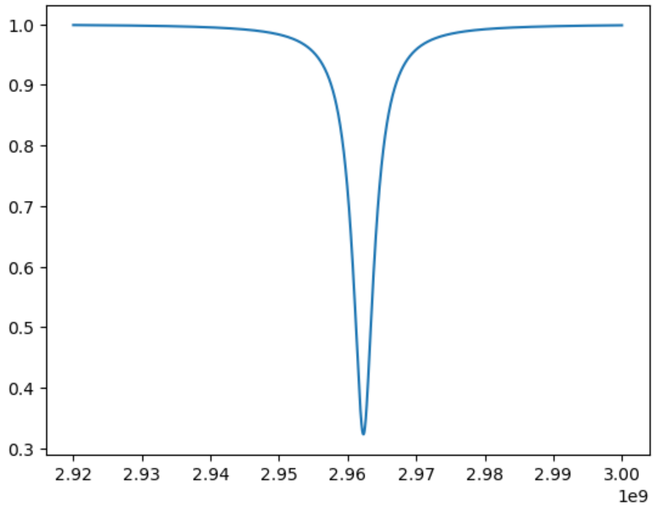


Figure 3: This plot shows the transmission coefficient $|S21|$ as a function of frequency for a single YIG resonator.

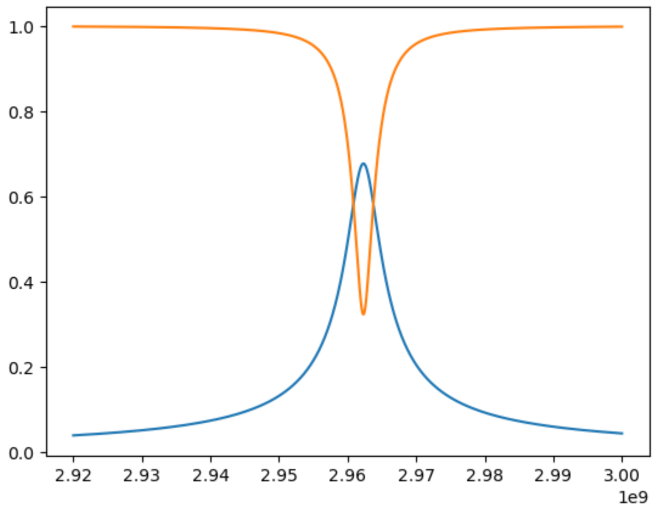


Figure 4: This plot compares the reflection coefficient $|S11|$ and the transmission coefficient $|S21|$ across frequencies.

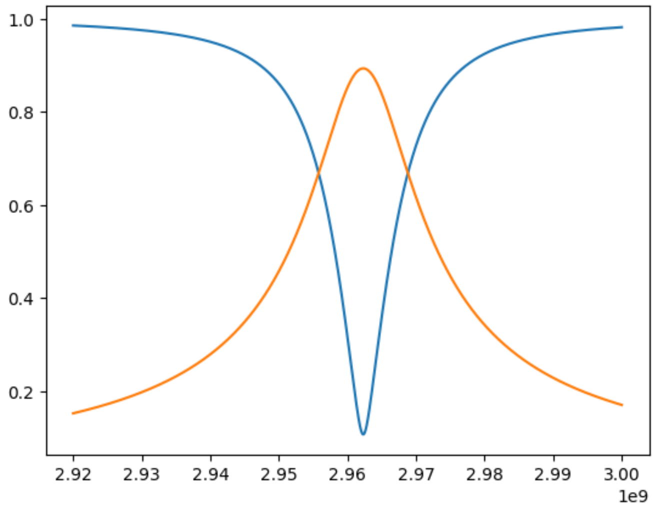


Figure 5: This plot shows the transmission coefficient $|S21|$ for YIG resonators spaced at $\lambda/2$

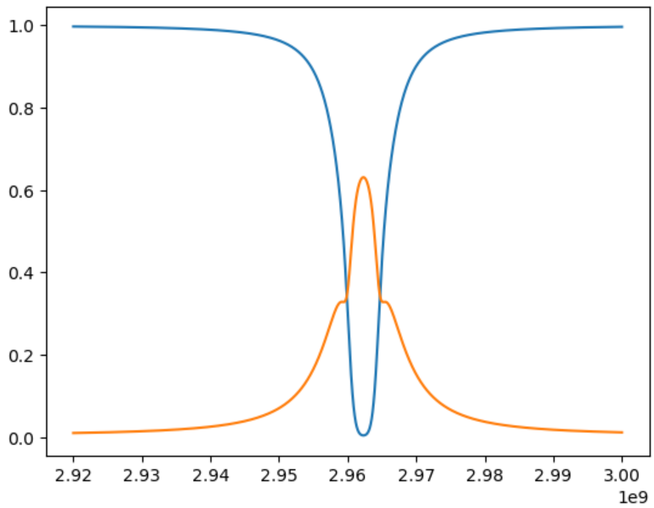


Figure 6: This plot shows the transmission coefficient $|S21|$ for YIG resonators spaced at $\lambda/4$

Equations Used

Hamiltonian for Qubit-Waveguide System

The Hamiltonian H describes the interaction between each qubit and the waveguide mode in the waveguide quantum electrodynamics (wQED) framework:

$$H = \frac{\omega_q}{2}\sigma_z + \int_{-\infty}^{\infty} \omega_k a_k^\dagger a_k \, dk + \int_{-\infty}^{\infty} g_k \left(\sigma_+ a_k + \sigma_- a_k^\dagger\right) dk$$

where:

- ω_q is the qubit's resonant frequency,
- σ_z is the Pauli operator for the qubit state,
- g_k represents the coupling strength between the qubit and the waveguide mode k,
- $a_k^\dagger$ and a_k are the photon creation and annihilation operators for the waveguide modes.

This Hamiltonian forms the basis for understanding photon-qubit interactions, enabling predictions about how qubits affect light transmission and reflection.

Transmission Coefficient S_{21}

The modified function for the transmission coefficient S_{21} is defined as follows:

$$S_{21} = 1 - \frac{\gamma_{\text{res}}}{2(\gamma_{\text{int}} + i(\omega - \omega_{\text{resonator}}))}$$

where:

– ω is the probe frequency,

– γ_{res} represents the decay rate due to coupling between the resonator and the waveguide,

– γ_{int} denotes the internal decay rate of the resonator,

– $\omega_{\text{resonator}}$ is the resonator's intrinsic frequency.

This equation allows calculation of the transmission coefficient, providing insight into the power transmitted through the system.

Reflection Coefficient S_{22}

The reflection coefficient S_{22} represents the portion of the signal that is reflected rather than transmitted:

$$S_{22} = S_{21} - 1$$

This coefficient helps analyze the power reflected back in the system.

Transfer Matrix Equation

The transfer matrix T describes how each resonator affects light propagation through the waveguide, modeling the relationship between incoming and outgoing wave amplitudes. It includes the reflection and transmission coefficients of the resonator:

$$T = \begin{pmatrix} \frac{1+2r}{1+r} & \frac{r}{1+r} \\ \frac{-r}{1+r} & \frac{1}{1+r} \end{pmatrix}$$

where:

– r is the reflection coefficient of the resonator,

– The elements T_{11}, T_{12}, T_{21}, and T_{22} determine the relationship between incident and transmitted/reflected waves.

This matrix allows us to model the scattering properties of each individual resonator in the array.

Propagation Phase Matrix

The propagation phase matrix P represents the phase shift experienced by light as it travels between adjacent resonators. This phase shift depends on the distance d between the resonators and the probe frequency ω_p:

$$P = \begin{pmatrix} e^{i\phi} & 0 \\ 0 & e^{-i\phi} \end{pmatrix}$$

where the phase shift ϕ is given by:

$$\phi = \frac{2\pi\omega_p \cdot d}{0.7\,c}$$

with:

– ω_p as the probe frequency,

– d as the distance between resonators, and

– c as the speed of light.

This matrix is used to account for the phase accumulated between resonators, crucial for controlling interference in Bragg and anti-Bragg configurations.

Total Transfer Matrix for Multi-Resonator Configurations

For an array with multiple resonators, the total transfer matrix T_{total} combines the effects of each individual resonator and phase shift. For a configuration with N resonators, this can be represented as a product of individual transfer matrices T_i and propagation phase matrices P_i:

$$T_{\text{total}} = T_1 \cdot P_1 \cdot T_2 \cdot P_2 \cdots T_N$$

This multiplication allows calculation of the cumulative transmission and reflection properties of the resonator array.

Normalized Distance Calculation

To interpret the spacing between resonators in terms of wavelength, we define a normalized distance d_{norm} as:

$$d_{\text{norm}} = \frac{d \times 2.9623 \times 10^9}{0.7\,c}$$

where:

– d represents the physical distance between resonators,

– c is the speed of light.

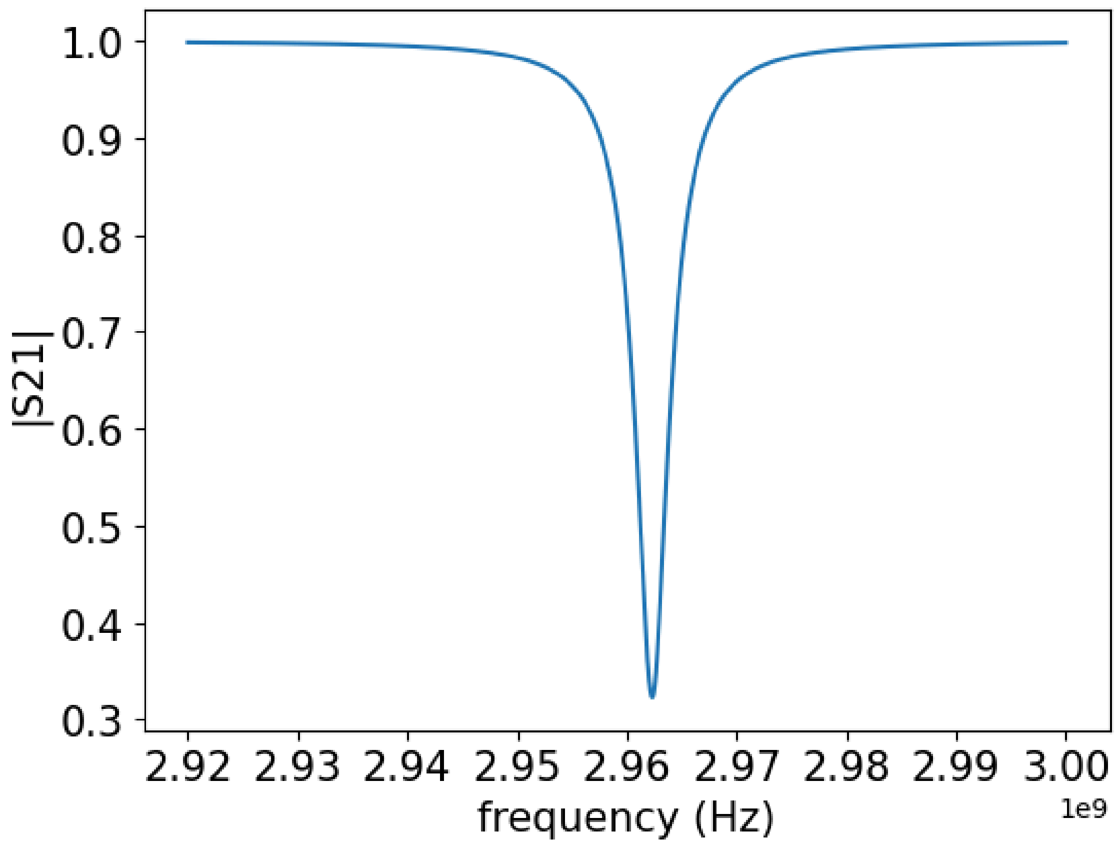
This normalized distance expresses spacing in terms of wavelength, aiding in the analysis of transmission properties across different configurations.

Code for Contour Plot Generation

In [1]:
```python
def S21(omega_p,kappa_r,gammar_r,omega_m_t):
    return 1-kappa_r/2/(gammar_r+1j*(omega_p-omega_m_t))
# scattering Parameter S21 - to describe the transmission coeff b/w 2 parameters
# omega_p - probe frequency
# kappa_r - resonator decay rate
# gammar_r - internal decay rate
# omega_m_t - frequency of resonato
# return Value - S21
```

In [2]:
```python
import numpy as np
import matplotlib.pyplot as plt
prob_freq=np.linspace(2.92e9,3.0e9,8001)

# define probe frequencies with range 8001 points

trans_coef=S21(prob_freq,3.306e6,2.442e6,2.9623e9)

# returns probe frequencies
```

In [3]:
```python
plt.plot(prob_freq,np.abs(trans_coef))
plt.xlabel('frequency (Hz)', fontsize=15)
plt.ylabel('|S21|', fontsize=15)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
# get and plot the magnitude of transmission coeff as a function of frequency
```



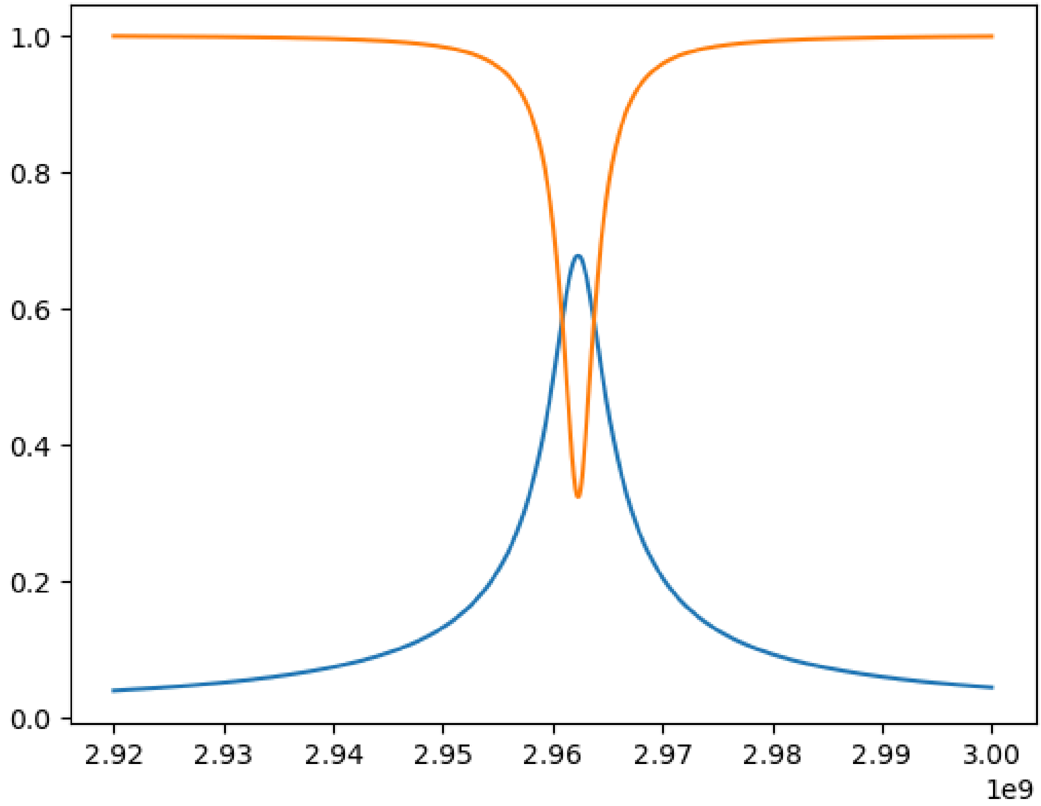
In [4]:
```
reflec_coef=trans_coef-1
# reflection coeff - S22 = S21 - 1
```

In [5]:
```
reflec_coef.shape
# to define relection coeff array
```

Out[5]:
```
(8001,)
```

In [6]:
```
plt.plot(prob_freq,np.abs(reflec_coef))
plt.plot(prob_freq,np.abs(trans_coef))
```

Out[6]:
```
[<matplotlib.lines.Line2D at 0x21c601320d0>]
```



In [6]:
```
from scipy.constants import c
def transfer_matrix(r):
    T11=(1+2*r)/(1+r)
    T12=r/(1+r)
    T21=-r/(1+r)
    T22=1/(1+r)
#     matrix=np.array([T11,T12],[T21,T22])
    return T11,T12,T21,T22

# define parameters of transfer matrix function based on reflection coeff r
# T11 - port 1 to port 1
# T12 - port 1 to port 2
# T21 - port 2 to port 1
# T22 - port 2 to port 2

def propgation_phase(prob_freq,distance):
    phi=prob_freq*distance*2*np.pi/(0.7*c)
    P11=np.exp(1j*phi)
```

```
    P12=0
    P21=0
    P22=np.exp(-1j*phi)
    return P11,P12,P21,P22,phi\

# phi - phase shift, 0.7c - modified speed of light
# *2pi - converts frequency to angular frequency
# P11 - phase shift for transmission from 1 to 1
# P12 - transmission from 1 to 2, set to 0 since no direct coupling
# P21 - transmission from 1 to 2, set to 0 since no direct coupling
# P22 - phase shift for transmission from 2 to 2 (reverse phase shift)
```

In [7]:
```
T11=np.array([],dtype=complex)
T12=np.array([],dtype=complex)
T21=np.array([],dtype=complex)
T22=np.array([],dtype=complex)

# values are initialized

for i,r in enumerate(reflec_coef):
    T11_1=(1+2*r)/(1+r)
    T12_1=r/(1+r)
    T21_1=-r/(1+r)
    T22_1=1/(1+r)
    T11=np.append(T11,T11_1)
    T12=np.append(T12,T12_1)
    T21=np.append(T21,T21_1)
    T22=np.append(T22,T22_1)

# loop through each r
# Caculate matrix values using r
# each loop is appended to the respective array
# creates a new array that includes the existing values plus the new value
```

In [9]:
```
T11.shape
# to determine the dimensions of the T11 array
```

Out[9]:
```
(8001,)
```

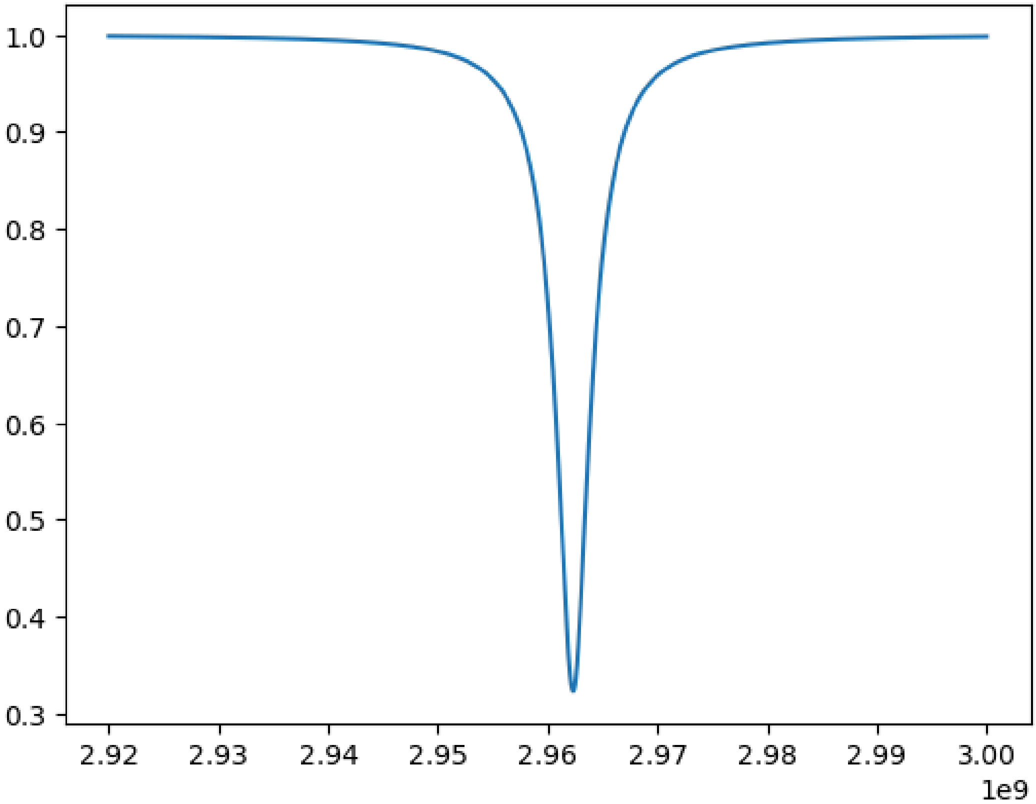
In [10]:
```
T_matrix_array = np.zeros((8001, 2, 2),dtype=complex)
for i in range(8001):
    T_matrix_array[i] = np.array([[T11[i], T12[i]], [T21[i], T22[i]]])
```

In [11]:
```
T_matrix_array.shape
# shape of the T matrix
```

Out[11]:
```
(8001, 2, 2)
```

In [12]:
```
plt.plot(prob_freq,np.abs(1/T_matrix_array[:,1,1]))

# plots magnitude of inverse T22 against probe frequenc
# helps us understand output-input relationship
```

Out[12]:
```
[<matplotlib.lines.Line2D at 0x17c1eb1a2d0>]
```



In [13]:
```
0.7*c/2.9623e9
```
Out[13]:
```
0.07084181905951456
```

In [14]:
```
2.9623e9*0.07084181905951456/(0.7*c)
```
Out[14]:
```
1.0
```

In [15]:
```
np.degrees(1.5707963267948963)
```
Out[15]:
```
89.99999999999999
```

In [16]:
```
np.exp(1j*1.5707963267948963)
```
Out[16]:
```
(2.83276944882399e-16+1j)
```

In [17]:
```
propgation_phase(2920100000.0,0.07084181905951456/2)[0]
# 3D array storing propagation phase matrices for each frequency
# lambda/ 2 - distance between 2 qubits
```
Out[17]:
```
(-0.9989987003030782+0.04473920867382864j)
```

In [12]:
```
#the distance between each YIG is lambda/2
propgation_phase_array=np.zeros((8001,2,2),dtype=complex)
propgation_phase_zero_imag=np.zeros((8001,2,2))
for i,case in enumerate(prob_freq):
    propgation_phase_array[i]=np.array([[propgation_phase(case,0.07084181905951456/2)[
# propgation_phase_array=propgation_phase_zero_real+1j*propgation_phase_zero_imag

# loop over each probe frequency
```

```
# create propogation phase matrix (2 by 2 matrix for each frequency)
# off-diagonal elements are set to 0 (no coupling)

# to make code more efficient since calling propgation_phase function twice is redunda
# for i, case in enumerate(prob_freq):
    # phase_result = propgation_phase(case, 0.07084181905951456 / 2)
    # propgation_phase_array[i] = np.array([[phase_result[0], 0],
                                           # [0, phase_result[3]]])
```

In [13]:
```
final_T_matrix=np.zeros((8001,2,2),dtype=complex)
final_T_matrix_imag=np.zeros((8001,2,2))
for i,case in enumerate(prob_freq):
    final_T_matrix[i]=T_matrix_array[i]@propgation_phase_array[i]\
    @T_matrix_array[i]@propgation_phase_array[i]\
    @T_matrix_array[i]@propgation_phase_array[i]\
    @T_matrix_array[i]
```

In [14]:
```
final_T_matrix.shape
```

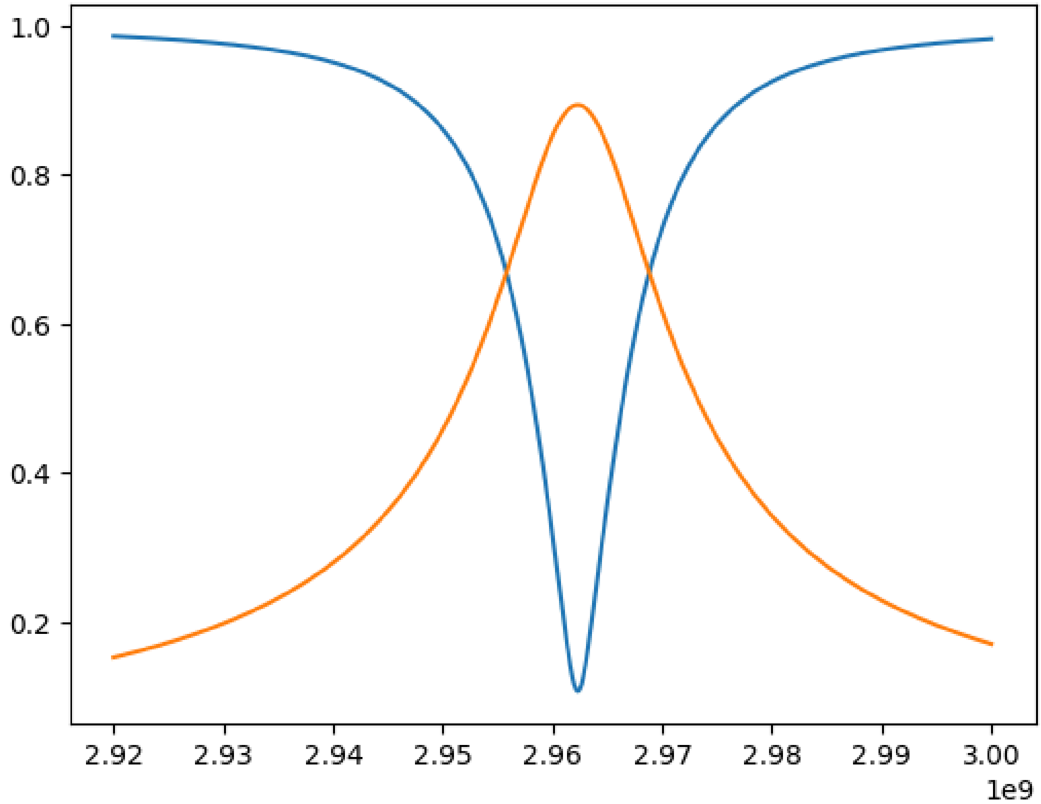
Out[14]:
```
(8001, 2, 2)
```

In [15]:
```
1/final_T_matrix[:,1,1]
```

Out[15]:
```
array([-0.94502461+0.27979638j, -0.94501715+0.2797979j ,
       -0.94500967+0.27979944j, ..., -0.94048424-0.28214288j,
       -0.94049677-0.28213386j, -0.94050927-0.28212487j])
```

In [16]:
```
plt.plot(prob_freq,np.abs(1/final_T_matrix[:,1,1]))
# plt.plot(prob_freq,np.angle(1/final_T_matrix[:,1,1]))
plt.plot(prob_freq,np.abs(final_T_matrix[:,0,1]/final_T_matrix[:,1,1]))
```

Out[16]:
```
[<matplotlib.lines.Line2D at 0x20d35a2a550>]
```



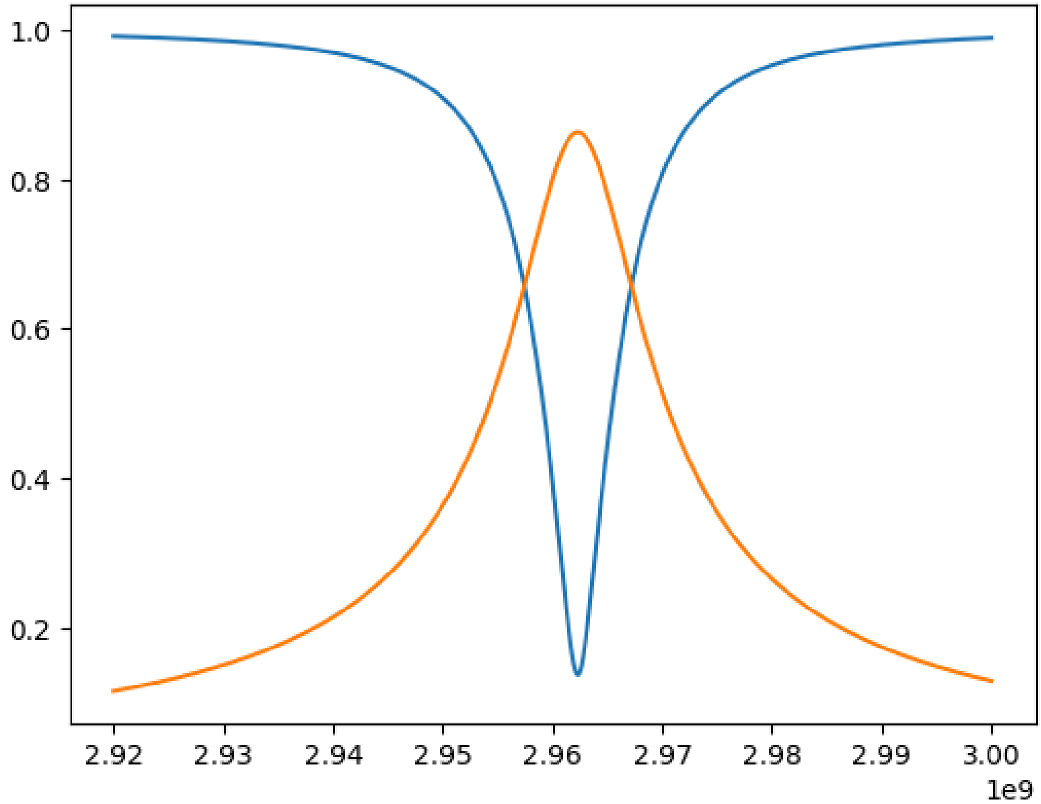
In [17]:
```
final_T_matrix=np.zeros((8001,2,2),dtype=complex)
final_T_matrix_imag=np.zeros((8001,2,2))
for i,case in enumerate(prob_freq):
    final_T_matrix[i]=T_matrix_array[i]@propgation_phase_array[i]\
    @T_matrix_array[i]@propgation_phase_array[i]\
    @T_matrix_array[i]
```

In [18]:
```
1/final_T_matrix[:,1,1]
```

Out[18]:
```
array([0.97030753-0.20235515j, 0.97030231-0.20235994j,
       0.97029708-0.20236475j, ..., 0.96723928+0.20593187j,
       0.96724755+0.20592085j, 0.96725582+0.20590985j])
```

In [19]:
```
plt.plot(prob_freq,np.abs(1/final_T_matrix[:,1,1]))
# plt.plot(prob_freq,np.angle(1/final_T_matrix[:,1,1]))
plt.plot(prob_freq,np.abs(final_T_matrix[:,0,1]/final_T_matrix[:,1,1]))
```

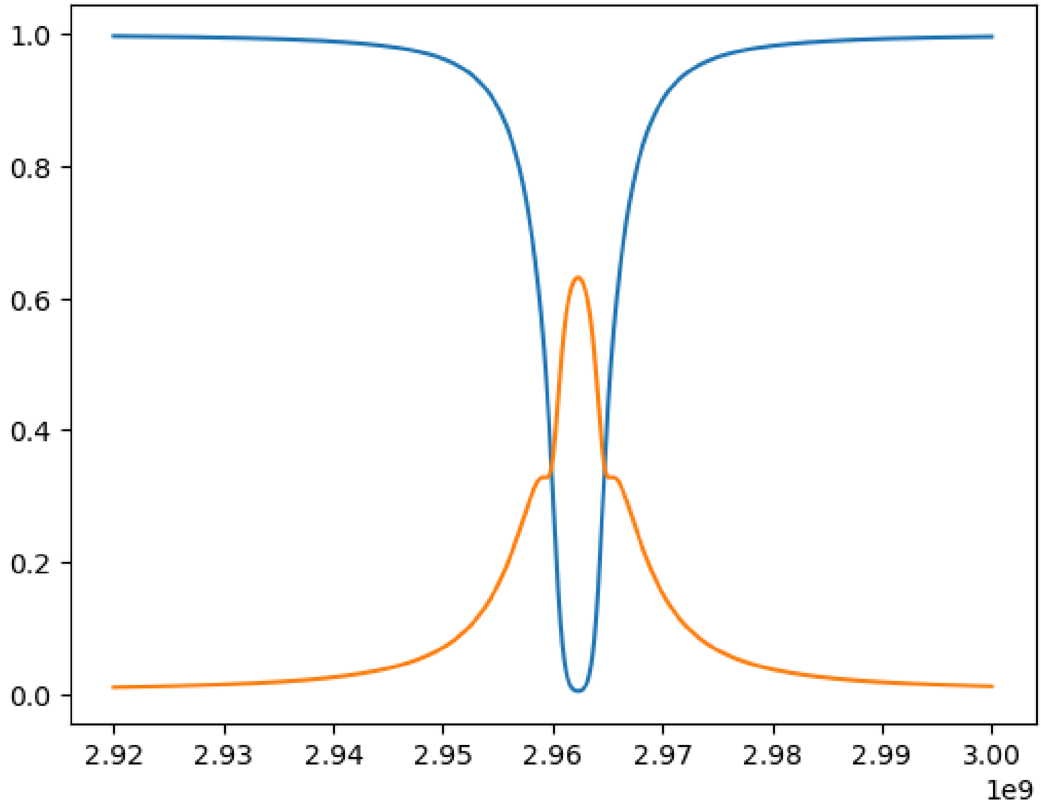
Out[19]:
```
[<matplotlib.lines.Line2D at 0x20d36a3f250>]
```



In [23]:
```
propgation_phase_array_2=np.zeros((8001,2,2),dtype=complex)

# initaialize and store 2nd set of prop phase matrices

for i,case in enumerate(prob_freq):
    propgation_phase_array_2[i]=np.array([[propgation_phase(case,0.07084181905951456/4

# construct 2 by 2 matrix like done previously but with distance lambda/4 b/w qubits

    final_T_matrix_2=np.zeros((8001,2,2),dtype=complex)

# initaialize and store final matrix calculated using 2nd set of prop phase matrix

for i,case in enumerate(prob_freq):
    final_T_matrix_2[i]=T_matrix_array[i]@propgation_phase_array_2[i]\
    @T_matrix_array[i]@propgation_phase_array_2[i]\
    @T_matrix_array[i]@propgation_phase_array_2[i]\
    @T_matrix_array[i]

# final transfer matrix for each frequency using array 2
# combine tranfer matrices and new prop phase matrices

# same reduncy as prev
```

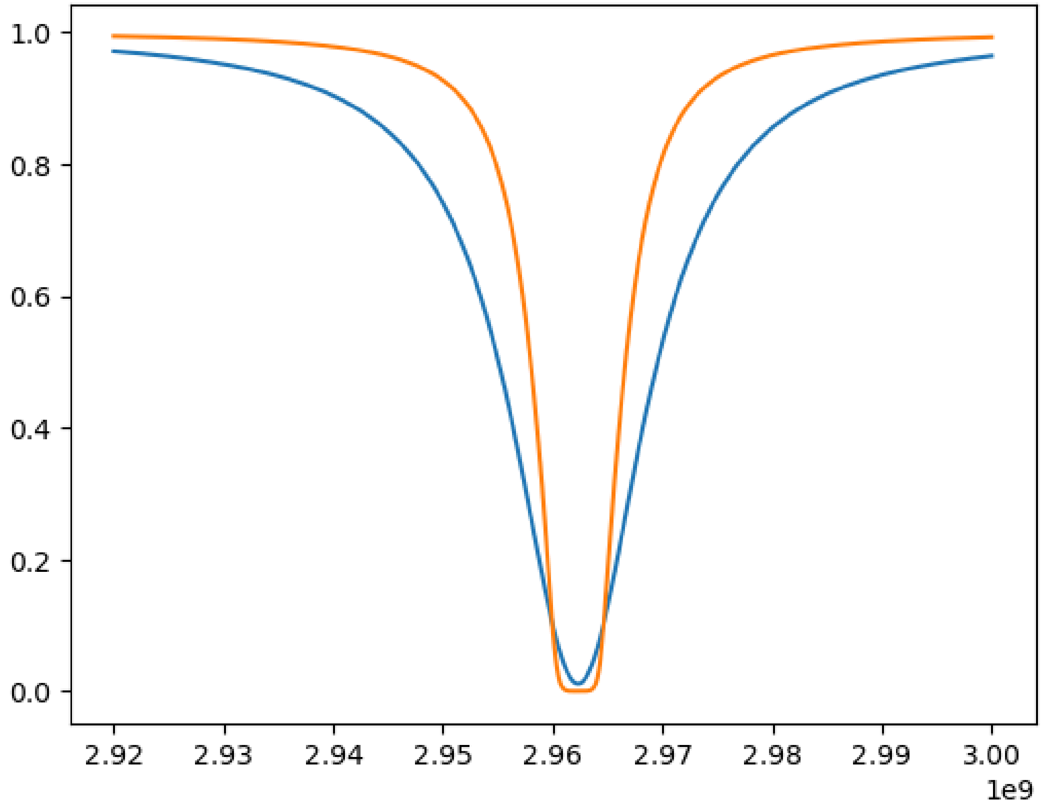
In [24]:
```
plt.plot(prob_freq,np.abs(1/final_T_matrix_2[:,1,1]))
plt.plot(prob_freq,np.abs(final_T_matrix_2[:,0,1]/final_T_matrix_2[:,1,1]))

# plotted against probe frequency, ratio of T12 to T22
```

Out[24]: [<matplotlib.lines.Line2D at 0x17c208d4550>]



In [25]:
```
plt.plot(prob_freq,np.abs(1/final_T_matrix[:,1,1])**2)
# plt.plot(prob_freq,np.abs(final_T_matrix[:,0,1]/final_T_matrix[:,1,1]))
plt.plot(prob_freq,np.abs(1/final_T_matrix_2[:,1,1])**2)
# plt.plot(prob_freq,np.abs(final_T_matrix_2[:,0,1]/final_T_matrix_2[:,1,1]))

# 1st - The result is plotted against prob_freq, showing how the squared magnitude of
# 2nd - Similar to the first plot, this line computes and plots the squared magnitude
```

Out[25]:
```
[<matplotlib.lines.Line2D at 0x17c20944810>]
```



In [26]:
```
def contour(x: np.array, y: np.array, z: np.array,
            xname: str = 'xname', yname: str = 'yname', zname: str = 'zname',
            show=False):
    """
    Plotting contour map.

    Parameters
    ----------
    x : np.array
        x-axis array.
    y : np.array
        y-axis array.
    z : np.array
        z-axis mesh array.
    xname : str, optional
        x-axis title. The default is 'xname'.
    yname : str, optional
        y-axis title. The default is 'yname'.
    zname : str, optional
        z-axis title. The default is 'zname'.

    """
    if np.iscomplexobj(z):
        z = np.abs(z)
    a = plt.pcolormesh(*np.meshgrid(x, y), z.transpose(), shading='auto')
    plt.xlabel(xname)
    plt.ylabel(yname)
    b = plt.colorbar(a)
    b.ax.set_ylabel(zname)
    if show:
        plt.show()
```

```
# the function checks if the z array contains complex numbers if yes, it converts z to
# creates coordinate grid for x and y values
# plots 2D color mesh, shading is calculated automatically
# added color bar set to z-axis
# if returns TRUE, plot is displayed
```

In [27]:

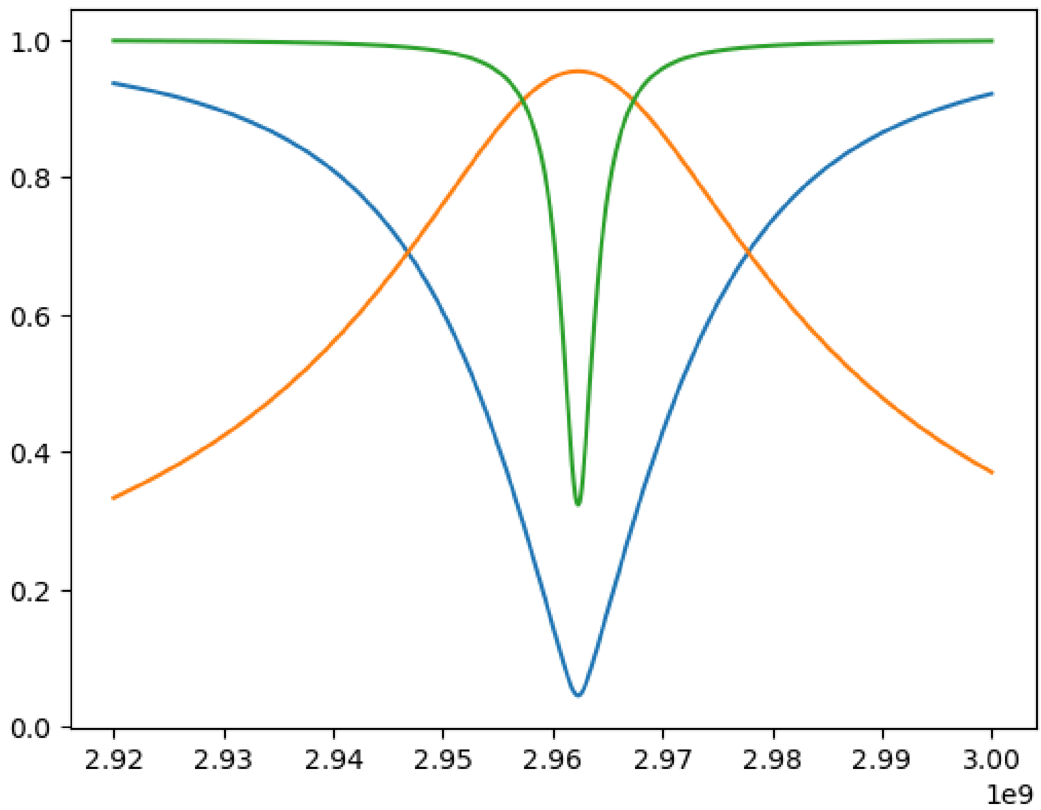
```
propgation_phase_array_3=np.zeros((8001,2,2),dtype=complex)
for i,case in enumerate(prob_freq):
    propgation_phase_array_3[i]=np.array([[propgation_phase(case,0.07084181905951456/2
final_T_matrix_3=np.zeros((8001,2,2),dtype=complex)
for i,case in enumerate(prob_freq):
    final_T_matrix_3[i]=T_matrix_array[i]@propgation_phase_array_3[i]\
    @T_matrix_array[i]@propgation_phase_array_3[i]\
    @T_matrix_array[i]@propgation_phase_array_3[i]\
    @T_matrix_array[i]@propgation_phase_array_3[i]\
    @T_matrix_array[i]@propgation_phase_array_3[i]\
    @T_matrix_array[i]@propgation_phase_array_3[i]\
    @T_matrix_array[i]@propgation_phase_array_3[i]\
    @T_matrix_array[i]@propgation_phase_array_3[i]\
    @T_matrix_array[i]@propgation_phase_array_3[i]\
    @T_matrix_array[i]
```

In [28]:

```
plt.plot(prob_freq,np.abs(1/final_T_matrix_3[:,1,1]))
plt.plot(prob_freq,np.abs(final_T_matrix_3[:,0,1]/final_T_matrix_3[:,1,1]))
plt.plot(prob_freq,np.abs(trans_coef))
```

Out[28]:

```
[<matplotlib.lines.Line2D at 0x17c209c7f10>]
```

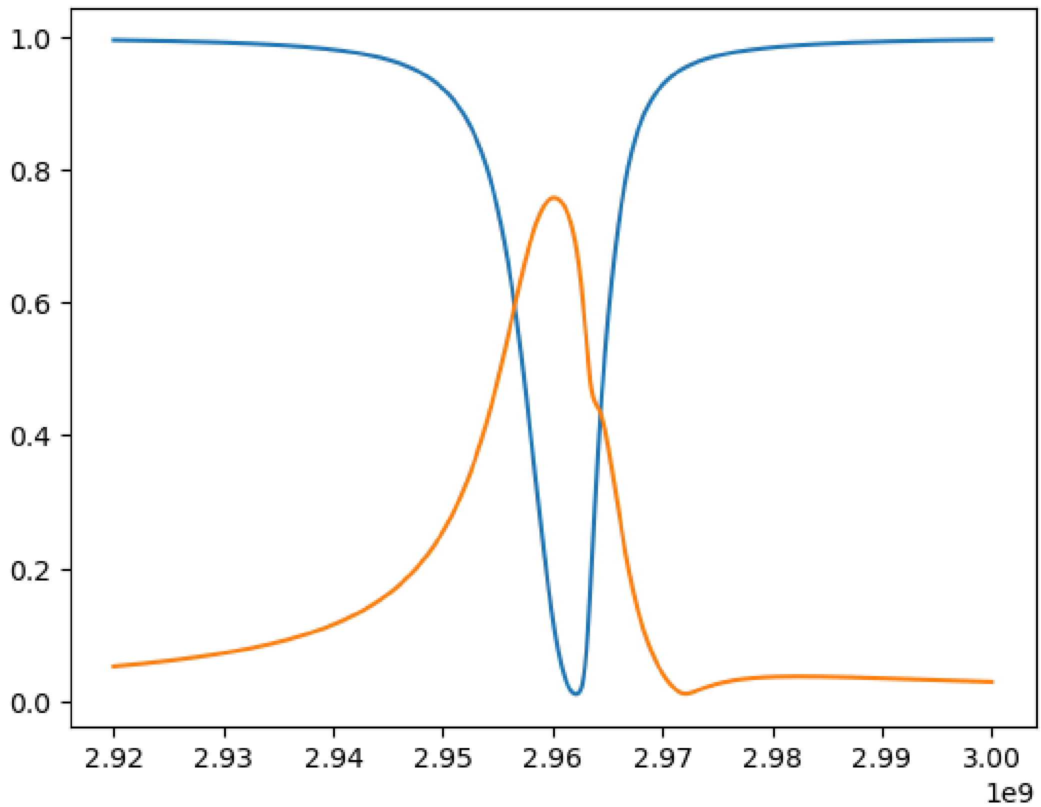


In [ ]:

In [29]:
```
propgation_phase_array_4=np.zeros((8001,2,2),dtype=complex)
for i,case in enumerate(prob_freq):
    propgation_phase_array_4[i]=np.array([[propgation_phase(case,0.07084181905951456/1
final_T_matrix_4=np.zeros((8001,2,2),dtype=complex)
for i,case in enumerate(prob_freq):
    final_T_matrix_4[i]=T_matrix_array[i]@propgation_phase_array_4[i]\
    @T_matrix_array[i]@propgation_phase_array_4[i]\
    @T_matrix_array[i]@propgation_phase_array_4[i]\
    @T_matrix_array[i]
```

In [30]:
```
plt.plot(prob_freq,np.abs(1/final_T_matrix_4[:,1,1]))
plt.plot(prob_freq,np.abs(final_T_matrix_4[:,0,1]/final_T_matrix_4[:,1,1]))
```

Out[30]: [<matplotlib.lines.Line2D at 0x17c20a2fd90>]



In [96]:
```
# 2YIG
trans_coef_2d_2YIG=np.array([],dtype=complex)
distance=np.linspace(0.0001,0.0501,101)
# distance is created as an array of 101 values ranging from 0.0001 to 0.0501

for i, case1 in enumerate(distance):
    i_str = str(i)
    variable_name0= 'two_d' + i_str
    variable_name1 = 'propgation_phase_array_' + i_str
    variable_name1=np.zeros((8001,2,2),dtype=complex)
# loop over each distance in array

    for i,case2 in enumerate(prob_freq):
        variable_name1[i]=np.array([[propgation_phase(case2,case1)[0],0],[0,propgatior
    variable_name2 = 'final_T_matrix_' + i_str
    variable_name2=np.zeros((8001,2,2),dtype=complex)
# Creaste prop phase array for current distance
```

```
    for i,case2 in enumerate(prob_freq):
        variable_name2[i]=T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]
# A 3D array variable_name2 is initialized to store the final transfer matrices for ea
    trans_coef_2d_2YIG_1=np.abs(1/variable_name2[:,1,1])
    trans_coef_2d_2YIG=np.append(trans_coef_2d_2YIG,trans_coef_2d_2YIG_1)

# The final transfer matrix is computed by multiplying the transfer matrix with the co
# for 2 YIG resonators
```

In [97]:

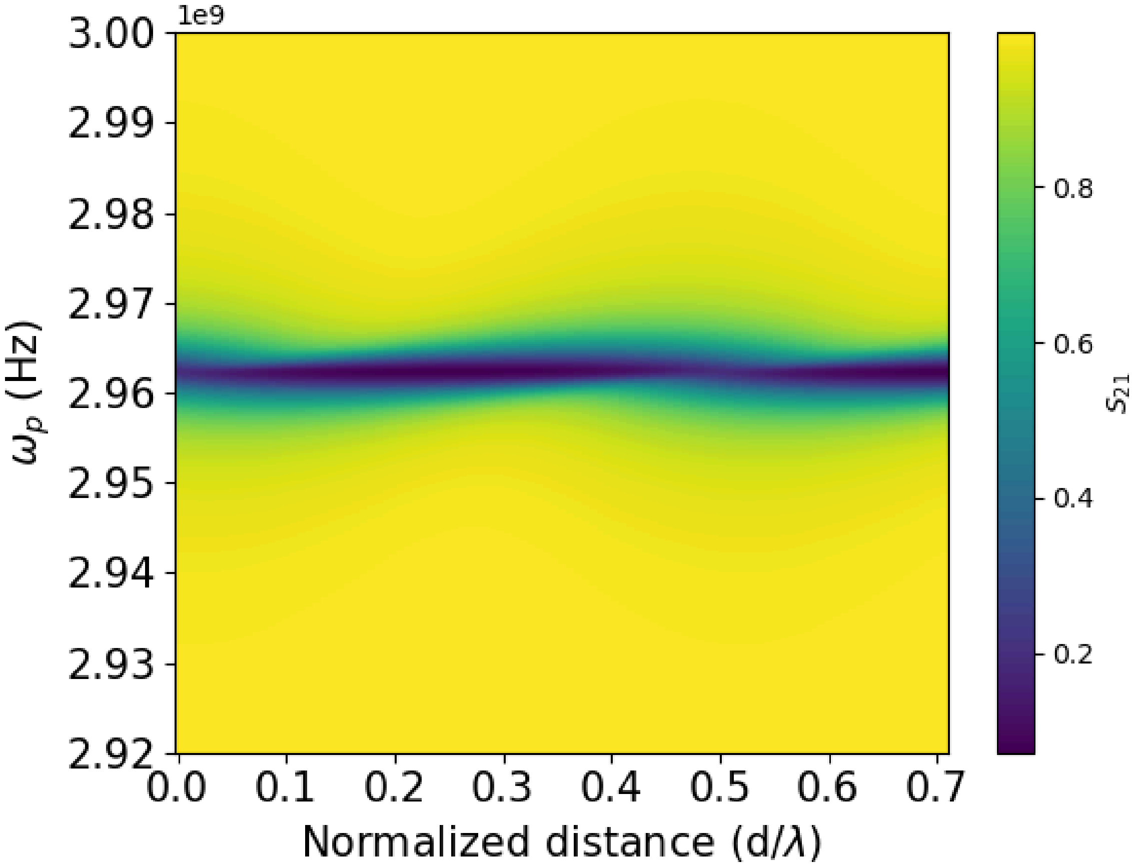
```
variable_name0=np.zeros((11,8001,2,2),dtype=complex)
# 4D array for complex matrices
```

In [99]:

```
trans_coef_2d_2YIG_reshape=trans_coef_2d_2YIG.reshape(101,8001)
normlized_dis=distance*2.9623e9/(0.7*c)
contour(normlized_dis,prob_freq,trans_coef_2d_2YIG_reshape,'Normalized distance (d/$\l
plt.xlabel('Normalized distance (d/$\lambda$)', fontsize=15)
plt.ylabel('$\omega_{p}$ (Hz)', fontsize=15)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)

#visualize tranfer coeff as contour plot
```

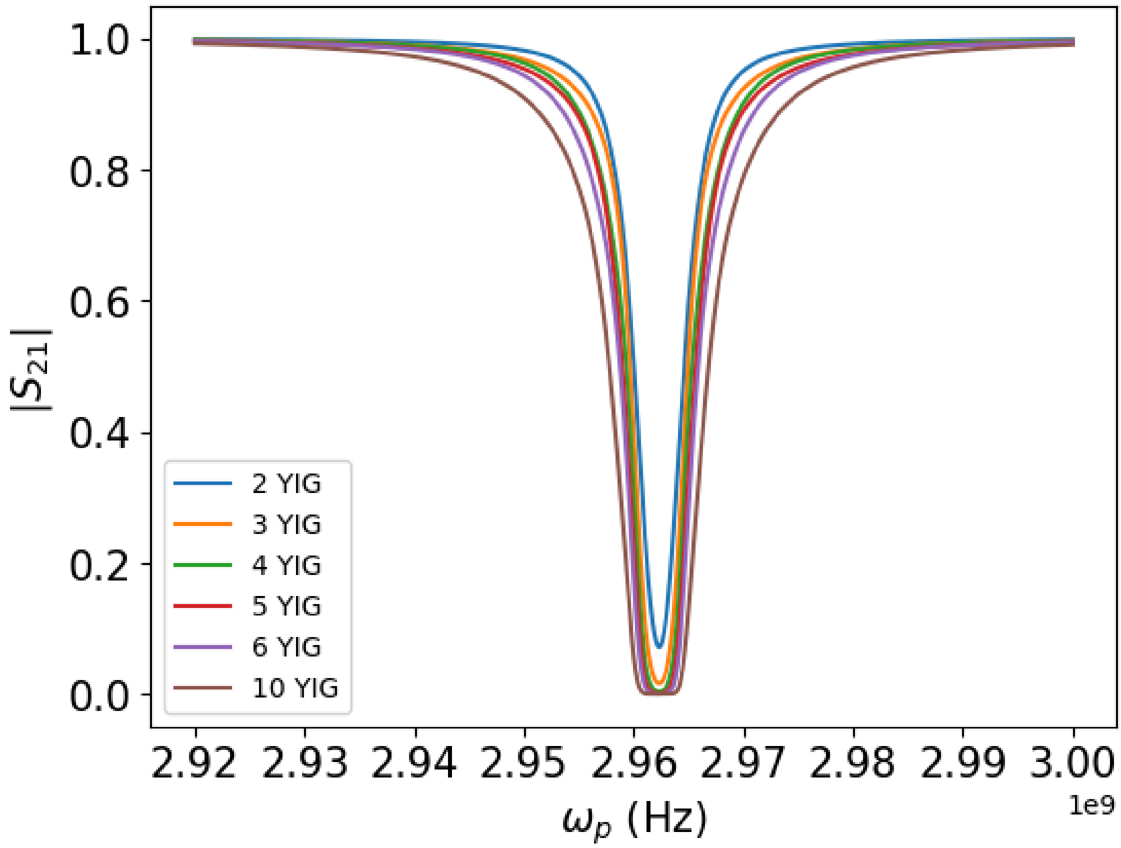
Out[99]:

```
(array([2.91e+09, 2.92e+09, 2.93e+09, 2.94e+09, 2.95e+09, 2.96e+09,
        2.97e+09, 2.98e+09, 2.99e+09, 3.00e+09, 3.01e+09]),
 [Text(0, 2910000000.0, '2.91'),
  Text(0, 2920000000.0, '2.92'),
  Text(0, 2930000000.0, '2.93'),
  Text(0, 2940000000.0, '2.94'),
  Text(0, 2950000000.0, '2.95'),
  Text(0, 2960000000.0, '2.96'),
  Text(0, 2970000000.0, '2.97'),
  Text(0, 2980000000.0, '2.98'),
  Text(0, 2990000000.0, '2.99'),
  Text(0, 3000000000.0, '3.00'),
  Text(0, 3010000000.0, '3.01')])
```



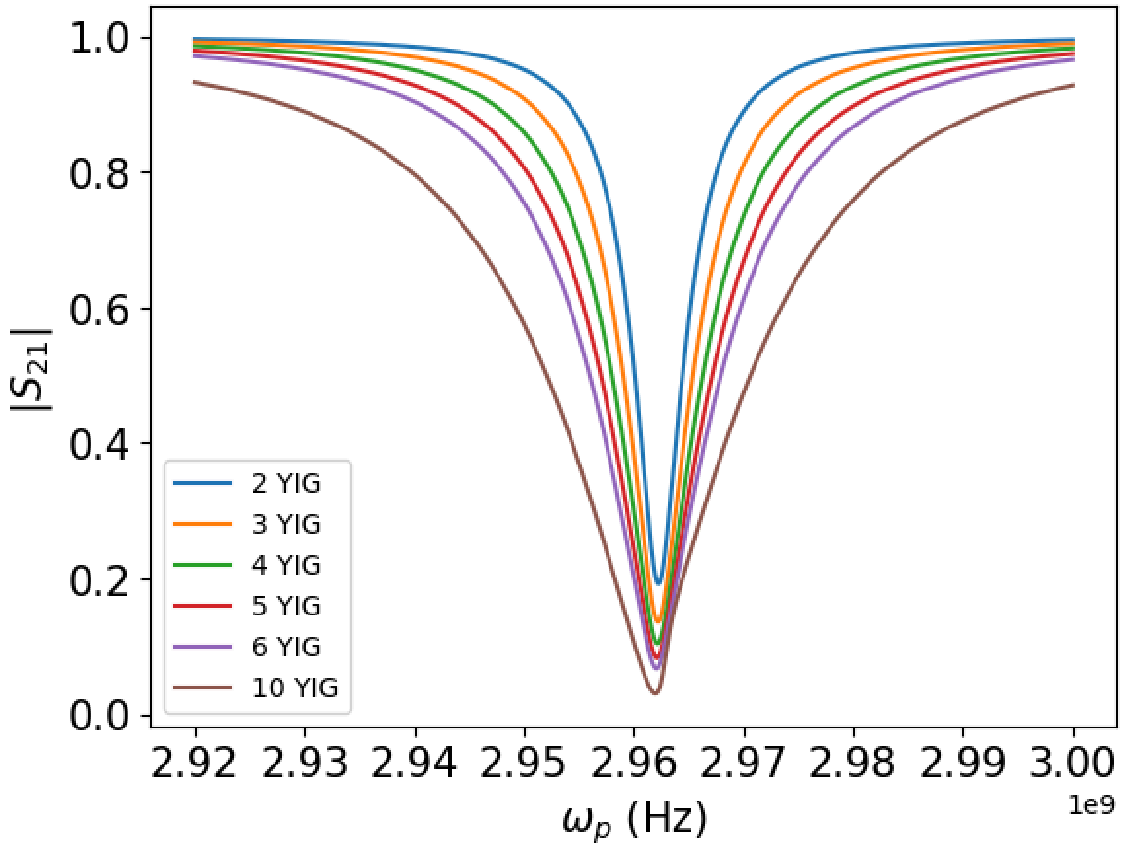
In [117…

```
plt.plot(prob_freq,np.abs(trans_coef_2d_2YIG_reshape[35,:]),label='2 YIG')
plt.plot(prob_freq,np.abs(trans_coef_2d_3YIG_reshape[35,:]),label='3 YIG')
plt.plot(prob_freq,np.abs(trans_coef_2d_4YIG_reshape[35,:]),label='4 YIG')
plt.plot(prob_freq,np.abs(trans_coef_2d_5YIG_reshape[35,:]),label='5 YIG')
plt.plot(prob_freq,np.abs(trans_coef_2d_6YIG_reshape[35,:]),label='6 YIG')
plt.plot(prob_freq,np.abs(trans_coef_2d_10YIG_reshape[35,:]),label='10 YIG')
plt.legend()
plt.xlabel('$\omega_{p}$ (Hz)', fontsize=15)
plt.ylabel('$|S_{21}|$ ', fontsize=15)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
```



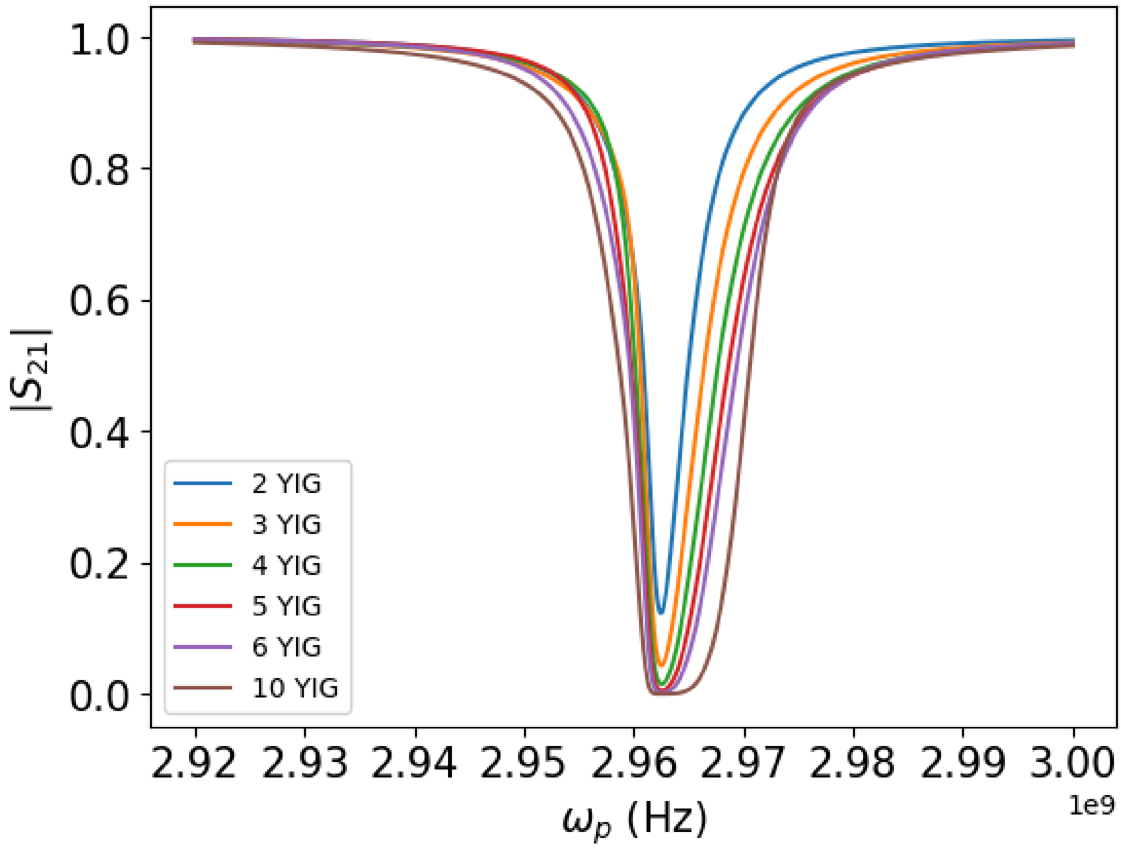
In [123…

```
plt.plot(prob_freq,np.abs(trans_coef_2d_2YIG_reshape[71,:]),label='2 YIG')
plt.plot(prob_freq,np.abs(trans_coef_2d_3YIG_reshape[71,:]),label='3 YIG')
plt.plot(prob_freq,np.abs(trans_coef_2d_4YIG_reshape[71,:]),label='4 YIG')
plt.plot(prob_freq,np.abs(trans_coef_2d_5YIG_reshape[71,:]),label='5 YIG')
plt.plot(prob_freq,np.abs(trans_coef_2d_6YIG_reshape[71,:]),label='6 YIG')
plt.plot(prob_freq,np.abs(trans_coef_2d_10YIG_reshape[71,:]),label='10 YIG')
# accesses the 35th row of the reshaped arrays, which corresponds to the specific dist
plt.legend()
plt.xlabel('$\omega_{p}$ (Hz)', fontsize=15)
plt.ylabel('$|S_{21}|$ ', fontsize=15)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()

# for each resonator configuration (2 YIG, 3 YIG, etc)
# plots the absolute values of transmission coefficients across all probe frequencies
```



In [126…

```
plt.plot(prob_freq,np.abs(trans_coef_2d_2YIG_reshape[60,:]),label='2 YIG')
plt.plot(prob_freq,np.abs(trans_coef_2d_3YIG_reshape[60,:]),label='3 YIG')
plt.plot(prob_freq,np.abs(trans_coef_2d_4YIG_reshape[60,:]),label='4 YIG')
plt.plot(prob_freq,np.abs(trans_coef_2d_5YIG_reshape[60,:]),label='5 YIG')
plt.plot(prob_freq,np.abs(trans_coef_2d_6YIG_reshape[60,:]),label='6 YIG')
plt.plot(prob_freq,np.abs(trans_coef_2d_10YIG_reshape[60,:]),label='10 YIG')
plt.legend()
plt.xlabel('$\omega_{p}$ (Hz)', fontsize=15)
plt.ylabel('$|S_{21}|$ ', fontsize=15)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
```



In [125…

```
normlized_dis[80]
# retrieve the value of normalized_dis at index 80
```

Out[125]:

```
0.566049835144857
```

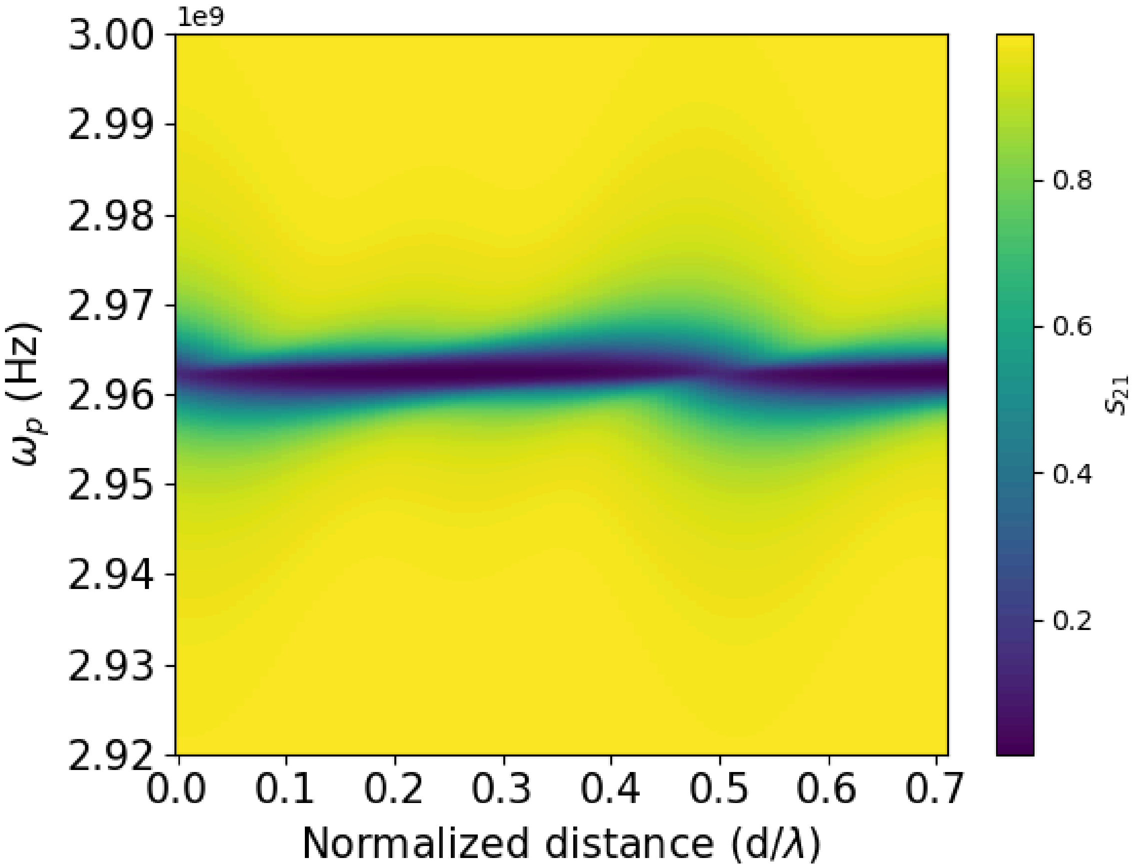
In [84]:

```
# 3YIG
trans_coef_2d_3YIG=np.array([],dtype=complex)
distance=np.linspace(0.0001,0.0501,101)
for i, case1 in enumerate(distance):
    i_str = str(i)
    variable_name0= 'two_d' + i_str
    variable_name1 = 'propgation_phase_array_' + i_str
    variable_name1=np.zeros((8001,2,2),dtype=complex)
    for i,case2 in enumerate(prob_freq):
        variable_name1[i]=np.array([[propgation_phase(case2,case1)[0],0],[0,propgatior
    variable_name2 = 'final_T_matrix_' + i_str
    variable_name2=np.zeros((8001,2,2),dtype=complex)
    for i,case2 in enumerate(prob_freq):
        variable_name2[i]=T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]
    trans_coef_2d_3YIG_1=np.abs(1/variable_name2[:,1,1])
    trans_coef_2d_3YIG=np.append(trans_coef_2d_3YIG,trans_coef_2d_3YIG_1)

# same as for 2 YIG resonators
```

In [85]:

```
trans_coef_2d_3YIG_reshape=trans_coef_2d_3YIG.reshape(101,8001)
normlized_dis=distance*2.9623e9/(0.7*c)
contour(normlized_dis,prob_freq,trans_coef_2d_3YIG_reshape,'Normalized distance (d/$\l
plt.xlabel('Normalized distance (d/$\lambda$)', fontsize=15)
```

```
plt.ylabel('$\omega_{p}$ (Hz)', fontsize=15)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
```



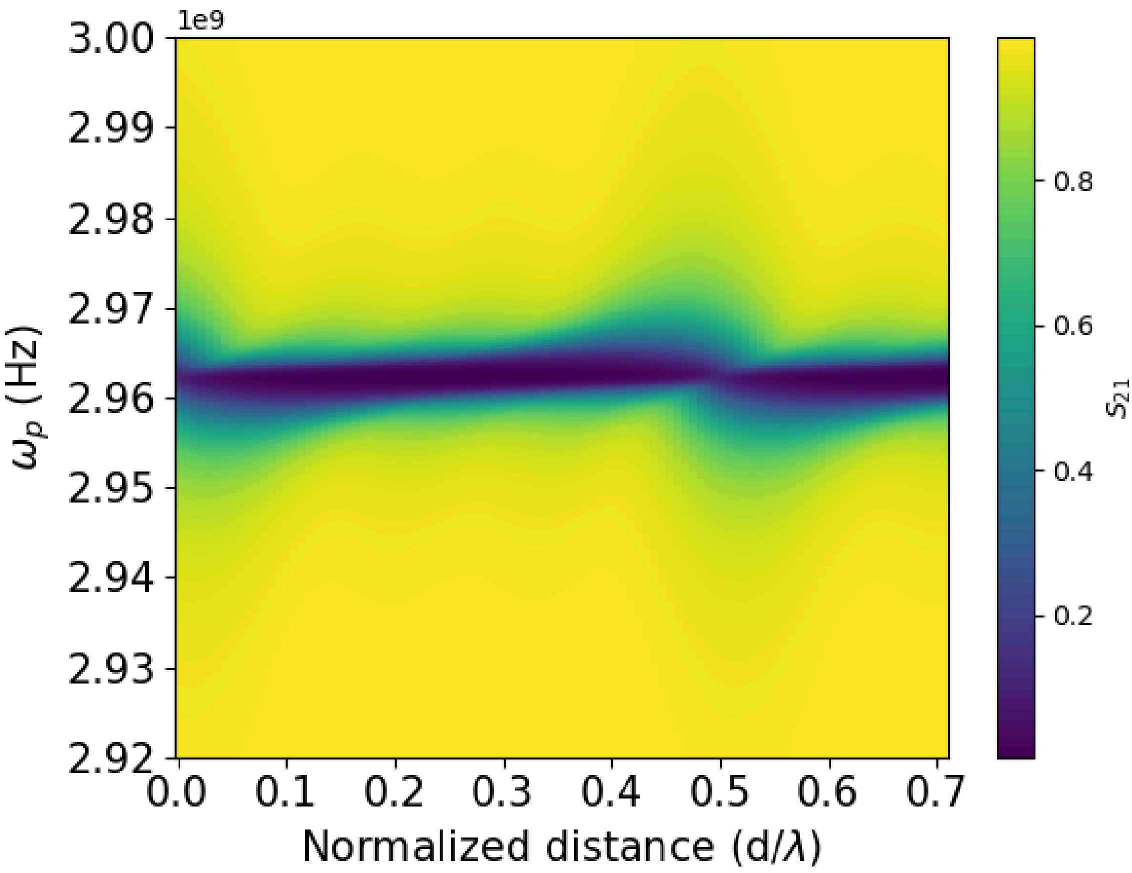
In [86]:

```
# 4YIG
trans_coef_2d_4YIG=np.array([],dtype=complex)
distance=np.linspace(0.0001,0.0501,101)
for i, case1 in enumerate(distance):
    i_str = str(i)
    variable_name0= 'two_d' + i_str
    variable_name1 = 'propgation_phase_array_' + i_str
    variable_name1=np.zeros((8001,2,2),dtype=complex)
    for i,case2 in enumerate(prob_freq):
        variable_name1[i]=np.array([[propgation_phase(case2,case1)[0],0],[0,propgatior
    variable_name2 = 'final_T_matrix_' + i_str
    variable_name2=np.zeros((8001,2,2),dtype=complex)
    for i,case2 in enumerate(prob_freq):
        variable_name2[i]=T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]
    trans_coef_2d_4YIG_1=np.abs(1/variable_name2[:,1,1])
    trans_coef_2d_4YIG=np.append(trans_coef_2d_4YIG,trans_coef_2d_4YIG_1)
```

In [87]:

```
trans_coef_2d_4YIG_reshape=trans_coef_2d_4YIG.reshape(101,8001)
normlized_dis=distance*2.9623e9/(0.7*c)
contour(normlized_dis,prob_freq,trans_coef_2d_4YIG_reshape,'Normalized distance (d/$\l
plt.xlabel('Normalized distance (d/$\lambda$)', fontsize=15)
plt.ylabel('$\omega_{p}$ (Hz)', fontsize=15)
```

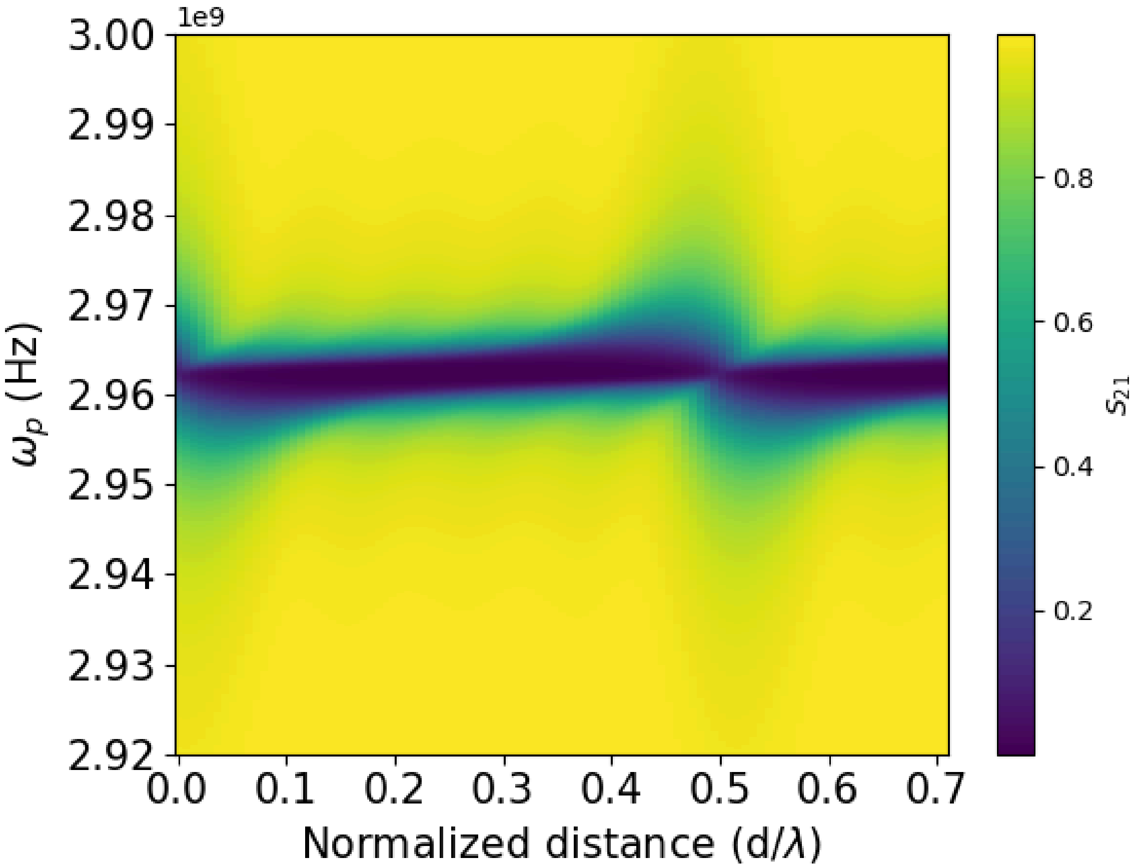
```
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
```



In [88]:
```
# 5YIG
trans_coef_2d_5YIG=np.array([],dtype=complex)
distance=np.linspace(0.0001,0.0501,101)
for i, case1 in enumerate(distance):
    i_str = str(i)
    variable_name0= 'two_d' + i_str
    variable_name1 = 'propgation_phase_array_' + i_str
    variable_name1=np.zeros((8001,2,2),dtype=complex)
    for i,case2 in enumerate(prob_freq):
        variable_name1[i]=np.array([[propgation_phase(case2,case1)[0],0],[0,propgation
    variable_name2 = 'final_T_matrix_' + i_str
    variable_name2=np.zeros((8001,2,2),dtype=complex)
    for i,case2 in enumerate(prob_freq):
        variable_name2[i]=T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]
    trans_coef_2d_5YIG_1=np.abs(1/variable_name2[:,1,1])
    trans_coef_2d_5YIG=np.append(trans_coef_2d_5YIG,trans_coef_2d_5YIG_1)
```

In [89]:
```
trans_coef_2d_5YIG_reshape=trans_coef_2d_5YIG.reshape(101,8001)
normlized_dis=distance*2.9623e9/(0.7*c)
contour(normlized_dis,prob_freq,trans_coef_2d_5YIG_reshape,'Normalized distance (d/$\l
plt.xlabel('Normalized distance (d/$\lambda$)', fontsize=15)
plt.ylabel('$\omega_{p}$ (Hz)', fontsize=15)
```

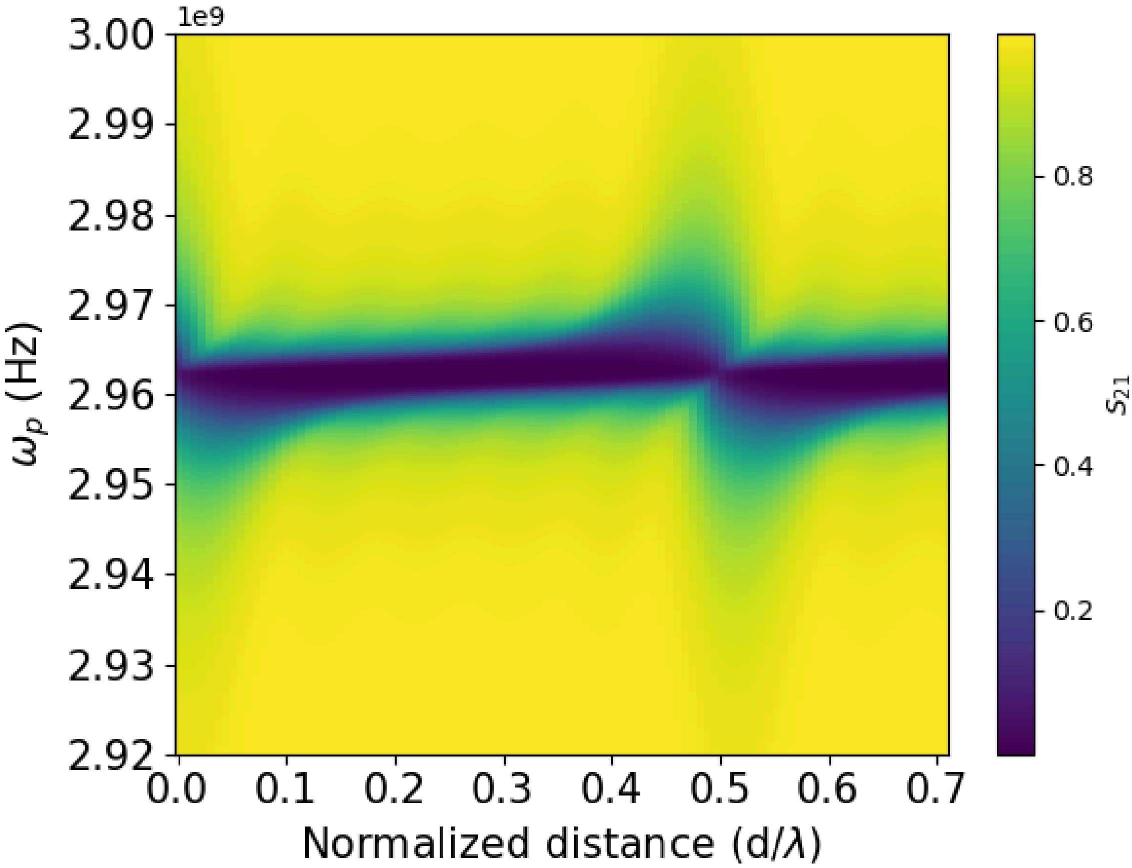
```
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
```



In [90]:
```
# 6YIG
trans_coef_2d_6YIG=np.array([],dtype=complex)
distance=np.linspace(0.0001,0.0501,101)
for i, case1 in enumerate(distance):
    i_str = str(i)
    variable_name0= 'two_d' + i_str
    variable_name1 = 'propgation_phase_array_' + i_str
    variable_name1=np.zeros((8001,2,2),dtype=complex)
    for i,case2 in enumerate(prob_freq):
        variable_name1[i]=np.array([[propgation_phase(case2,case1)[0],0],[0,propgatior
    variable_name2 = 'final_T_matrix_' + i_str
    variable_name2=np.zeros((8001,2,2),dtype=complex)
    for i,case2 in enumerate(prob_freq):
        variable_name2[i]=T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]
    trans_coef_2d_6YIG_1=np.abs(1/variable_name2[:,1,1])
    trans_coef_2d_6YIG=np.append(trans_coef_2d_6YIG,trans_coef_2d_6YIG_1)
```

In [91]:
```
trans_coef_2d_6YIG_reshape=trans_coef_2d_6YIG.reshape(101,8001)
normlized_dis=distance*2.9623e9/(0.7*c)
contour(normlized_dis,prob_freq,trans_coef_2d_6YIG_reshape,'Normalized distance (d/$\l
plt.xlabel('Normalized distance (d/$\lambda$)', fontsize=15)
```

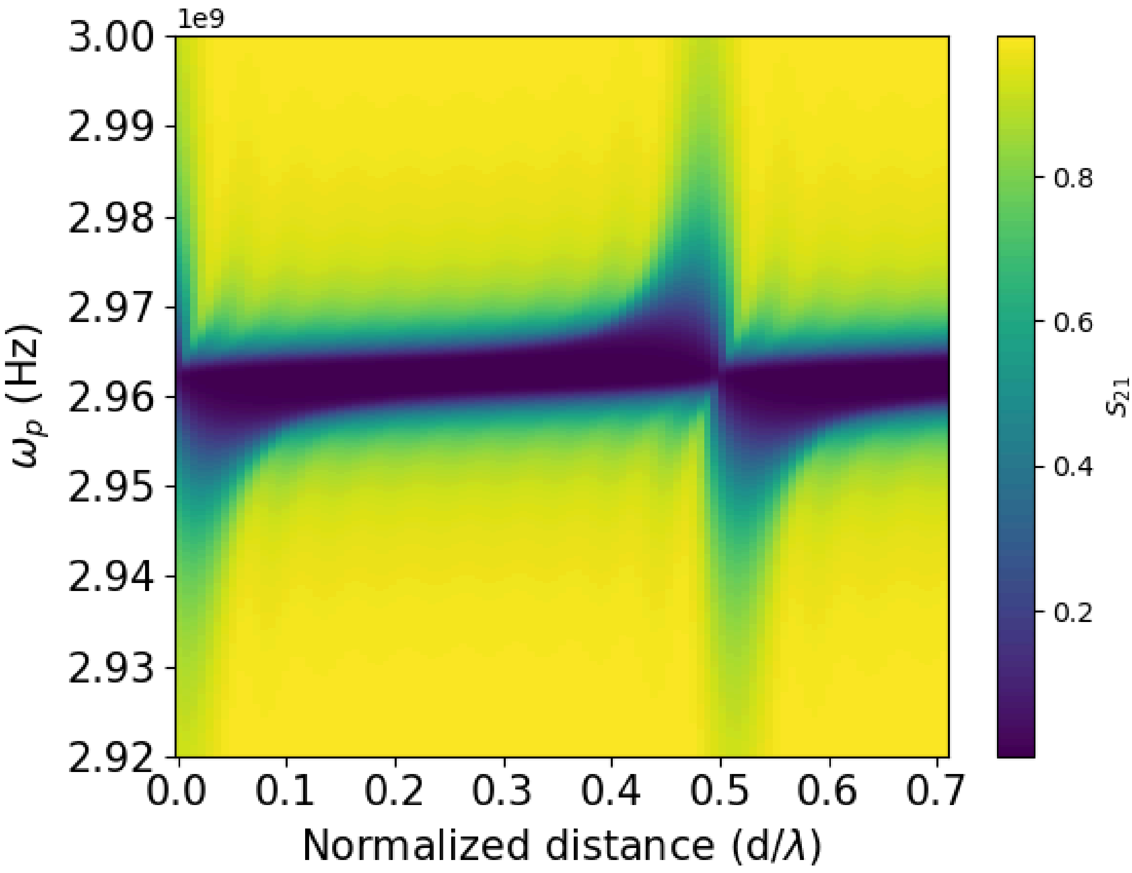
```
plt.ylabel('$\omega_{p}$ (Hz)', fontsize=15)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
```



In [92]:

```
# 10YIG
trans_coef_2d_10YIG=np.array([],dtype=complex)
distance=np.linspace(0.0001,0.0501,101)
for i, case1 in enumerate(distance):
    i_str = str(i)
    variable_name0= 'two_d' + i_str
    variable_name1 = 'propgation_phase_array_' + i_str
    variable_name1=np.zeros((8001,2,2),dtype=complex)
    for i,case2 in enumerate(prob_freq):
        variable_name1[i]=np.array([[propgation_phase(case2,case1)[0],0],[0,propgatior
    variable_name2 = 'final_T_matrix_' + i_str
    variable_name2=np.zeros((8001,2,2),dtype=complex)
    for i,case2 in enumerate(prob_freq):
        variable_name2[i]=T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]
    trans_coef_2d_10YIG_1=np.abs(1/variable_name2[:,1,1])
    trans_coef_2d_10YIG=np.append(trans_coef_2d_10YIG,trans_coef_2d_10YIG_1)
```

In [93]:
```
trans_coef_2d_10YIG_reshape=trans_coef_2d_10YIG.reshape(101,8001)
normlized_dis=distance*2.9623e9/(0.7*c)
contour(normlized_dis,prob_freq,trans_coef_2d_10YIG_reshape,'Normalized distance (d/$\
plt.xlabel('Normalized distance (d/$\lambda$)', fontsize=15)
plt.ylabel('$\omega_{p}$ (Hz)', fontsize=15)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
```



In [129…
```
# 10YIG_fine
trans_coef_2d_10YIG=np.array([],dtype=complex)
distance=np.linspace(0.0001,0.0501,1001)
for i, case1 in enumerate(distance):
    i_str = str(i)
    variable_name0= 'two_d' + i_str
    variable_name1 = 'propgation_phase_array_' + i_str
    variable_name1=np.zeros((8001,2,2),dtype=complex)
    for i,case2 in enumerate(prob_freq):
        variable_name1[i]=np.array([[propgation_phase(case2,case1)[0],0],[0,propgation
    variable_name2 = 'final_T_matrix_' + i_str
    variable_name2=np.zeros((8001,2,2),dtype=complex)
    for i,case2 in enumerate(prob_freq):
        variable_name2[i]=T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
```

```
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]
    trans_coef_2d_10YIG_1=np.abs(1/variable_name2[:,1,1])**2
    trans_coef_2d_10YIG=np.append(trans_coef_2d_10YIG,trans_coef_2d_10YIG_1)
```

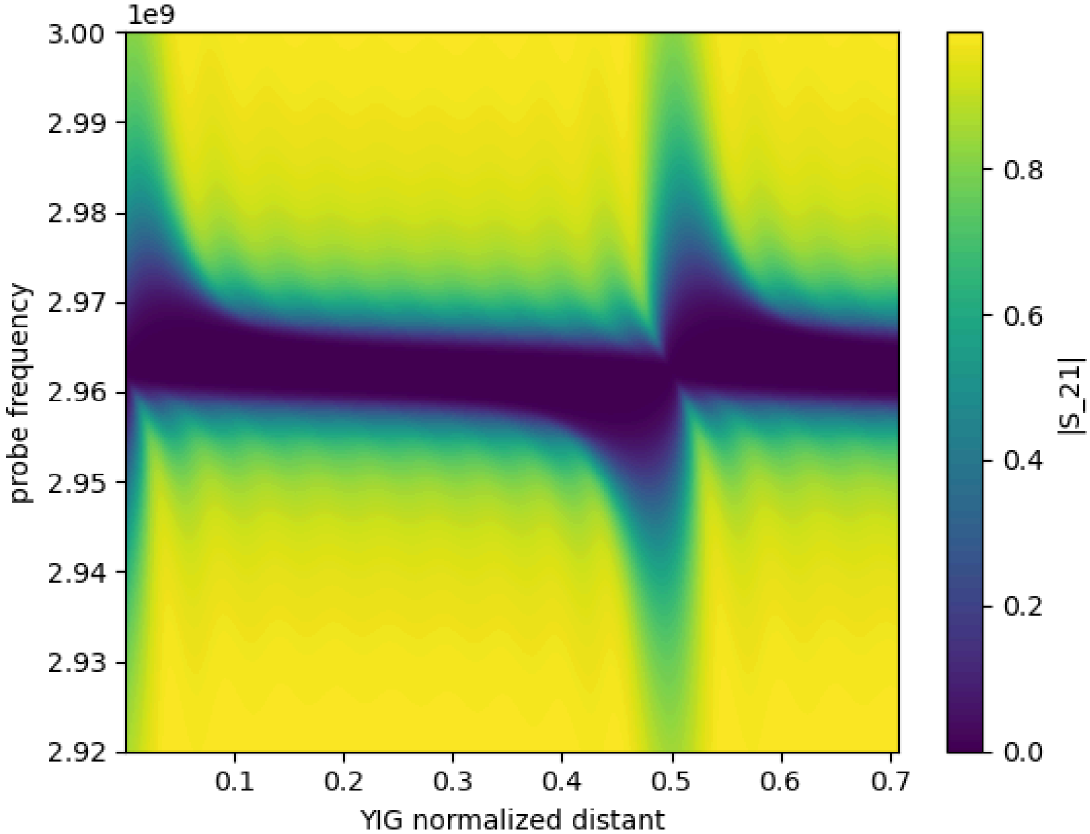
In [2]:

```
trans_coef_2d_10YIG_reshape=trans_coef_2d_10YIG.reshape(1001,8001)
normlized_dis=distance*2.9623e9/(0.7*c)
contour(normlized_dis,prob_freq,trans_coef_2d_10YIG_reshape)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[2], line 1
----> 1 trans_coef_2d_10YIG_reshape=trans_coef_2d_10YIG.reshape(1001,8001)
      2 normlized_dis=distance*2.9623e9/(0.7*c)
      3 contour(normlized_dis,prob_freq,trans_coef_2d_10YIG_reshape)

NameError: name 'trans_coef_2d_10YIG' is not defined
```

In [137…

```
trans_coef_2d_10YIG_reshape=trans_coef_2d_10YIG.reshape(1001,8001)
normlized_dis=distance*2.9623e9/(0.7*c)
contour(normlized_dis,prob_freq,trans_coef_2d_10YIG_reshape,
        xname = 'YIG normalized distant', yname= 'probe frequency', zname = '|S_21|')
```



In [139…

```
# 5YIG_fine
trans_coef_2d_10YIG=np.array([],dtype=complex)
distance=np.linspace(0.0001,0.0501,1001)
for i, case1 in enumerate(distance):
    i_str = str(i)
    variable_name0= 'two_d' + i_str
    variable_name1 = 'propgation_phase_array_' + i_str
    variable_name1=np.zeros((8001,2,2),dtype=complex)
```

```
    for i,case2 in enumerate(prob_freq):
        variable_name1[i]=np.array([[propgation_phase(case2,case1)[0],0],[0,propgatior
    variable_name2 = 'final_T_matrix_' + i_str
    variable_name2=np.zeros((8001,2,2),dtype=complex)
    for i,case2 in enumerate(prob_freq):
        variable_name2[i]=T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]@variable_name1[i]\
        @T_matrix_array[i]
    trans_coef_2d_10YIG_1=np.abs(1/variable_name2[:,1,1])**2
    trans_coef_2d_10YIG=np.append(trans_coef_2d_10YIG,trans_coef_2d_10YIG_1)
```

In [ ]:

```
trans_coef_2d_10YIG_reshape=trans_coef_2d_10YIG.reshape(1001,8001)
normlized_dis=distance*2.9623e9/(0.7*c)
contour(normlized_dis,prob_freq,trans_coef_2d_10YIG_reshape,
        xname = 'YIG normalized distant', yname= 'probe frequency', zname = '|S_21|')
```

Glossary of Terms

– **Superconducting Qubit**: A quantum bit based on superconducting circuits, which can maintain quantum states for long periods (long coherence times) due to their negligible resistance. Superconducting qubits are widely used in quantum computing and communication for their strong interactions with electromagnetic fields.

– **Waveguide Quantum Electrodynamics (wQED)**: A field of study focused on the interactions between quantum systems (like qubits or resonators) and the continuous spectrum of electromagnetic modes in a waveguide. Unlike cavity QED, which confines light to a single mode in a small cavity, wQED allows for broader light-matter interactions across multiple modes, enhancing quantum effects.

– **Cavity Quantum Electrodynamics (cavity QED)**: A branch of quantum electrodynamics that studies the interactions between light and matter within a confined cavity. This setup restricts the system to a single electromagnetic mode, which simplifies the study of light-matter interactions compared to waveguides.

– **Hamiltonian**: An operator in quantum mechanics that represents the total energy of a system, including kinetic and potential energies. In wQED, the Hamiltonian describes the energy interactions between the qubit and the photon modes within the waveguide.

– **Pauli Operator σ_z**: A mathematical operator used in quantum mechanics to represent the spin states of qubits. In the context of superconducting qubits, Pauli operators help model the qubit's energy levels and their interactions with electromagnetic fields.

– **Photon-Qubit Coupling**: The interaction between a photon and a qubit in a quantum system. This coupling enables the transfer of energy between the two, which can lead to quantum phenomena like resonance fluorescence and interference effects.

– **Resonance Fluorescence**: The process by which a qubit or atom absorbs and re-emits a photon at its resonance frequency. This interaction between the qubit and light results in a specific, detectable emission that provides insight into the system's energy levels.

– **Bragg Structure**: An arrangement of resonators or qubits at periodic intervals that creates constructive interference in the light they scatter, forming band gaps where certain wavelengths are blocked. Bragg structures are useful for creating reflective surfaces that prevent specific frequencies from passing through.

– **Anti-Bragg Structure**: An arrangement of resonators or qubits that creates **destructive interference**, allowing more light to pass through the waveguide. Anti-Bragg structures form transparency windows in the band gap, enabling selective transmission of certain wavelengths.

– **Photonic Band Gap**: A frequency range in which light propagation is suppressed due to interference effects in a periodic structure, such as a Bragg array of qubits or resonators. Photonic band gaps are used in photonic devices to control the transmission and reflection of specific wavelengths.

– **Yttrium Iron Garnet (YIG) Sphere**: A ferrimagnetic material often used as a resonator in quantum experiments due to its long coherence times, strong photon-magnon coupling, and low magnetic damping. YIG spheres enable experiments in photon-magnon coupling, where light interacts with magnons (spin waves) in the material.

– **Magnon**: A quantized spin-wave excitation in a magnetic material, representing the collective oscillation of electron spins. In YIG spheres, magnons can couple with photons, enabling studies in magnon-photon interactions within wQED setups.

– **Photon-Magnon Coupling**: The interaction between photons and magnons in a magnetic material, such as a YIG sphere. This coupling enables the study of light-matter interactions in multi-level systems, where energy can transfer between photons and magnons, affecting the system's resonance characteristics.

– **Magnetic Tunability**: The ability to control the resonance frequency of a system by applying an external magnetic field. In YIG spheres, this tunability adjusts the magnon's frequency, allowing researchers to align the magnon's resonance with the waveguide photons.

– **Transfer Matrix**: A mathematical representation of the relationship between incoming and outgoing electromagnetic fields in a resonator system. In wQED, the transfer matrix describes the transmission and reflection properties of each resonator or qubit.

– **Transmission Coefficient**: A measure of the transmitted signal's strength between two points in a waveguide, often represented by the $|S_{21}|$ scattering parameter. It quantifies how much of the input signal passes through the resonator array.

– **Reflection Coefficient**: A measure of the reflected signal strength at an input port, represented by the $|S_{11}|$ scattering parameter. It indicates how much of the input signal is reflected back from the resonator array.

– **Propagation Phase**: The phase shift that occurs as light travels through a medium or waveguide. In wQED, the propagation phase between resonators affects the constructive or destructive interference, which influences the formation of Bragg and anti-Bragg gaps.

– **Low Damping**: A property of a material, such as YIG, where energy loss during oscillations is minimal. Low damping is beneficial for maintaining stable oscillations (or quantum states) over extended periods, enhancing coherence in quantum experiments.

– **Multi-Level System**: A quantum system with more than two energy levels, allowing for more complex interactions and transitions. YIG spheres act as multi-level systems, in contrast to typical two-level qubits, which makes them useful for studying interactions beyond simple binary states.

– **Low-Power Regime**: An experimental condition where photon interactions occur with minimal power, often to avoid saturation or nonlinear effects. In this context, it allows for precise control over photon-magnon or photon-qubit interactions without excessive energy input.

– **Waveguide Mode**: A specific electromagnetic field distribution or pattern that can propagate within a waveguide. Each mode corresponds to a particular spatial configuration and frequency that satisfies the boundary conditions of the waveguide. Modes are characterized by how energy flows through the waveguide and are essential in defining how qubits or resonators interact with electromagnetic fields within the system.

– **Photon Creation Operator ($a_k^\dagger$)**: An operator in quantum mechanics used to add (or create) a photon in a specific quantum state or mode k within a system. Applying this operator increases the number of photons in that mode by one, and it is a fundamental tool for describing light-matter interactions in quantum electrodynamics.

– **Photon Annihilation Operator (a_k)**: An operator that removes (or annihilates) a photon from a specific quantum state or mode k. When applied to a quantum state with one or more photons, it decreases the photon count in that mode by one. This operator is essential in modeling the emission or absorption of photons in systems such as waveguides coupled with qubits or resonators.